

Kooperative Optimierung von Raumluchtströmungen mittels agentengestützter Regelungstechnik in einer Computational Steering Umgebung

Von der
Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig

zur Erlangung des Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von
Torsten Fahrig
aus Freiburg

Eingereicht am	05. Juli 2007
Mündliche Prüfung am	25. Oktober 2007

Berichterstatter	Prof. Dr. Manfred Krafczyk Prof. Dr. Uwe Rüppel
------------------	--

2007

Zusammenfassung

Die Planung eines Bauwerks im konstruktiven Ingenieurbau erfordert zur Vermeidung von Konstruktionsfehlern eine enge Kooperation zwischen den beteiligten Fachplanern. Die verbreitete bauplanorientierte Konstruktion benötigt bei fachübergreifenden Vorhaben eine Extraktion und Übertragung von Teilmodellen der einzelnen Fachdomänen. Ein zielführendes Zusammenwirken sowie nachhaltige Qualitätskontrollen im Bauwesen sind dadurch erschwert.

Im Rahmen dieser Arbeit wird der Ansatz verfolgt, durch eine kooperative und interaktive Simulation von Entwurfs- und Konstruktionsvarianten der einzelnen Fachmodelle mit gleichzeitiger Visualisierung ihrer Zielfunktionswerte über einen gemeinsamen virtuellen Entwurfsraum eine gegenseitige Verständigung mittelbar zu ermöglichen und zu verbessern. Durch die schrittweise Modifizierung der einzelnen Fachmodelle, welche in einem IFC Produktmodell vorgehalten werden, ist die interaktive Optimierung des Modells möglich. Partielle Verbindungen und Abhängigkeiten zwischen den fachmodellspezifischen Objekten finden in Klassen- und Eigenschaftsdefinitionen ihren Ausdruck.

Der virtuelle Entwurfsraum wird als Softwarelösung realisiert. Hierbei erfolgt eine Kopplung von Softwaremodulen zur dreidimensionalen Modellierung, Simulation und Visualisierung. Diese sind zur Ermöglichung eines interaktiven Prozessablaufs über automatisierte Schnittstellen verknüpft. Da die Konstruktionsbestandteile primär geometrische Eigenschaften besitzen, wird ein erweiterungsfähiges CAD-System zur Modellierung eingesetzt. Durch die Nutzung eines CFD-Kerns ist eine Optimierung von Raumluftrömungen unter Berücksichtigung energetischer und klimakomfortabhängiger Aspekte möglich. Über eine Multi-User-Umgebung kann ein ortsunabhängiger, vernetztkooperativer Zugriff der Fachplaner auf den virtuellen Entwurfsraum erfolgen.

Interaktive Elemente des Entwurfsraums wurden durch die Verwendung des Computational Steering Ansatzes unter Einbezug einer Virtual Reality Umgebung sowie einer Großrechenanlage realisiert. Das Computational Steering stellt Mechanismen zur Integration von Simulation, Datenanalyse, Visualisierung und Nachbearbeitung zur Verfügung und gewährt dem Benutzer die interaktive Kontrolle über ein laufendes Programm.

Eine Herausforderung bei der interaktiven Simulation stellt der notwendige Modelltransfer dar. Aus dem als Datenbasis eingesetzten IFC-Produktmodell wird für den Simulationskern ein interpretierbares Berechnungsmodell abgeleitet. Für die Strömungssimulation ist die Vorhaltung eines volumenorientierten Geometriemodells mit Informationen zum Rechengebiet sowie bauteilspezifische und globale Randbedingungen für die Simulation notwendig. Dieser Modelltransfer erfolgt in einem automatisierten Hintergrundprozess.

Der vorgestellte Software-Prototyp ermöglicht die praxisnahe Konzeption und Optimierung von HVAC-Systemen. Hierbei werden mess-, steuer- und regeltechnische Prozesse (MSR-Technik) einer Klimaanlageinstallation emuliert und über virtuelle geometrische Repräsentanten mit dem Entwurfsraum gekoppelt. Ausgehend von sensorisch zu erfassenden Größen der Raumtemperatur, Luftfeuchtigkeit und Luftgeschwindigkeit werden Flussstärke, Temperatur und Einfallswinkel der einströmenden Luft aus den Klimamodulen dynamisch angepasst. Die Emulation der HVAC-Systeme erfolgt durch den Einsatz von Software-Agenten.

Abstract

Functional Design of complex CAE objects in civil engineering requires a close co-operation of specialists in various disciplines. Especially in the early design phase, the effort to correct errors or inadequacies is relatively small. As the design goals of the different engineering disciplines often tend to be mutually exclusive, efficient communication between the engineering specialists is essential for an effective design process and mandatory for an economic optimization during the design and the economic lifetime of the construction. An automated computer-assisted system serving as a framework for this process would presuppose a consistent theoretical description of all partial models as part of an object and process oriented generalized theory about the interactions of all building components and processes. Such a theory is not yet available.

This thesis describes a heuristic strategy: A group of engineers from different disciplines come together in an interactive and virtual design space and evaluate in close cooperation the mutual implications of their specific design variants. In the course of a step-by-step modification of the partial models which are condensed into a single IFC product model, the interactive optimization of this model is potential. Partial connections and dependencies between the specific expert objects are defined in class- and property-definitions.

The virtual design space is developed as a software-prototype. This prototype is based on several linked modules for three-dimensional modeling, simulation and visualization. These modules are linked by automated interfaces to allow an interactive process flow. The basis for the modeler is an extendable CAD-system for the efficient handling of the geometry. By using a CFD-kernel the user can optimize the airflow within a room with respect to the human wellness index and the energy consumption of HVAC-installations. The virtual design space may externally be modified by additional engineers. This is possible by utilizing a multi-user-environment. Common software allows a remote sharing of PC GUI over the intranet or internet.

Interactive elements of the design space are introduced by utilizing a computational steering system, a virtual reality environment and a pc-cluster as compute server. Computational steering combines mechanisms for the integration of simulation, data analysis, visualization and adjustment to provide the user of an interactive control of the running program. The challenge of the data management concept of our prototype is the required model transfer. Thus the IFC product model data basis must be transformed into an interpretable calculation model for the CFD simulation. For the simulation process, product model specific data are being mapped to geometric objects. The physical attributes of all objects and the global boundary conditions relevant for the CFD simulation are extracted and transferred automatically by a daemon process.

Today common building simulation systems usually compute stationary states of HVAC systems with constant airflow rates of inlets. The dynamical behavior of HVAC installations depending on the climatic conditions inside a room which is controlled by measuring-, operation- and control- techniques cannot be modeled in such a framework. Yet, for a realistic modeling of HVAC engineering, such features are essential. To extend the present approach of HVAC simulation, we combine a Computational Steering approach and a software agent system. In our prototype, these controllers are emulated by interactive software-agents, which are bidirectionally coupled with the simulation kernel and wrap the HVAC objects of the modeler.

Vorwort und Danksagung

Die hier vorgestellte Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für rechnergestützte Modellierung im Bauingenieurwesen (iRMB), vormals Institut für Computeranwendungen im Bauingenieurwesen (CAB) an der Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften an der Technischen Universität Carolo-Wilhelmina zu Braunschweig.

Ich danke der DFG für die Förderung meines Projekts „Ein Prototyp für verteilte, interaktiv-kooperative Simulationen zur Beschleunigung von Entwurfszyklen im konstruktiven Ingenieurbau“ im Rahmen des DFG-Schwerpunktprogramms 1103 „Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau“.

Mein Dank gilt allen, die zu dem Gelingen dieser Arbeit beigetragen haben. Insbesondere möchte ich die Unterstützung folgender Personen hervorheben:

- Mein besonderer Dank gilt meinem Doktorvater Prof. Dr. Manfred Krafczyk für die Annahme und Betreuung meiner Dissertation sowie die Betreuung des DFG-Projekts. Er zeigte in den vergangenen vier Jahren stetiges Interesse und Engagement an diesem Projekt und ließ mir viele Freiheiten zur Verwirklichung eigener Ideen.
- Prof. Dr. Uwe Rüppel danke ich für die Übernahme des Zweitgutachtens.
- Ich danke Dr. Jonas Tölke, welcher mir bei komplizierten Sachverhalten immer mit Rat und Tat bei der Betreuung dieses Projektes zur Seite stand.
- Ich danke meinem Zimmerkollegen Bjørn Nachtwey mit dem ich gemeinsam dieses Projekt bearbeitet habe. Durch seine stetige Diskussions- und Hilfsbereitschaft konnten wir die Schnittstellen unserer beiden Arbeitsfelder schnell und effizient verbinden.
- Ich danke meinen Kollegen Sören Freudiger und Sebastian Geller, welche mich bei Programmierfragen zu Qt und C++ sowie bei diversen Fehlersuchen in Quellcode unterstützt haben.
- Ich danke meinen beiden studentischen Hilfskräften Sebastian Bindick und Lin Zhuang. Sebastian ist nun mein Nachfolger für die Bearbeitung des Projektes und unterstützte mich zusammen mit Lin bei der Programmierung von Modulen für den Software-Prototyp. Dadurch konnte die nun vorhandene Funktionsvielfalt erst erreicht werden.
- Ich danke Herrn Karl Tessmar für die Einführung in die Mess-, Steuer-, und Regeltechnik. Seine praxisorientierten Hinweise zur Auslegung von Klimaanlage erwiesen sich als außerordentlich hilfreich bei der Entwicklung der Software-Agenten.
- Ich danke meinem Vater Prof. Dr. Rudolf Fahrig, meiner Mutter Brigitte Fahrig und meiner Frau Karima Fahrig für die Korrekturen von Wort und Bild dieser Arbeit sowie die Geduld, die sie mir während der Erstellung gezeigt haben.

Torsten Fahrig, im Juni 2007

Inhaltsverzeichnis

Kapitel 1	Einführung	1
1.1	Problemstellung	1
1.2	Zielsetzung der Arbeit	4
1.3	Aufbau des Software Prototyps	6
1.4	Gliederung der Arbeit	8
Kapitel 2	Produktmodelle	9
2.1	Einleitung	9
2.2	Das STEP Produktmodell	12
2.3	Das IFC Produktmodell	14
2.3.1	Aufbau und Struktur	16
2.3.2	Entwicklung	20
2.4	Das Autodesk AEC Produktmodell	21
2.5	Schlussfolgerung und Ausblick	23
Kapitel 3	Der virtuelle Entwurfsraum	25
3.1	Der ADT als Modellierer	25
3.2	Schnittstellen des ADT	25
3.2.1	ObjectARX	26
3.2.2	Das Object Modeling Framework	27
3.3	Funktionen des virtuellen Entwurfsraums	32
3.3.1	Ereigniserkennung	36
3.3.2	Automatische Korrekturfunktionen	37
3.3.3	Pre-Processing	39
3.3.4	Automatische Erstellung von Bauplänen	44
3.4	Implementierung der Software-Module	47
3.5	Schlussfolgerung und Ausblick	48
Kapitel 4	Das HVAC-Agenten Framework	49
4.1	Einleitung	49
4.2	Thermische Behaglichkeit	50
4.3	Regelungstechnik	53
4.3.1	Regelungstechnik in der Praxis	53
4.3.2	Einführung in die Regelungstechnik	55
4.4	Software Agenten	58
4.5	Umsetzung	60
4.5.1	Funktionsweise der Agenten	63
4.5.2	Eingliederung in das Computational Steering Konzept	64
4.5.3	Anbindung des Agenten Frameworks an das ADT	65

4.6	Klimaregelung über HVAC-Agenten	66
4.6.1	Übertragungsverhalten	69
4.6.2	Interaktive Optimierung eines Reglers	71
4.6.3	Klimaregelung über ein Behaglichkeitsfeld	72
4.7	Ausblick	73
Kapitel 5	Eingesetzte Technologien des Software Prototyps	76
5.1	Virtual Reality	76
5.1.1	Stereoprojektion.....	76
5.1.2	VR-Technologie.....	77
5.1.3	Einsatzgebiete	79
5.2	Computational Steering.....	80
5.3	Vernetzt-kooperative Planung.....	81
Kapitel 6	Modelltransfer	83
6.1	Einleitung	83
6.2	Dreidimensionale geometrische Modellierung.....	84
6.3	Überblick	91
6.4	Schritt I: Vom IFC Modell zum AEC Modell	94
6.4.1	Konvertierung der IFC-Geometrie.....	94
6.4.2	Konvertierung bauteilspezifischer Metainformationen.....	96
6.5	Schritt II: Vom AEC Modell zum Facettenmodell.....	100
6.6	Schritt III: Vom Facettenmodell zum Oktalbaum	101
6.6.1	Der triangle-box overlap Test.....	102
6.6.2	Intelligentes Lokalisierungsverfahren	103
6.6.3	Benchmarks	105
6.7	Schritt IV: Vom Oktalbaum zum Voxelmmodell.....	106
6.8	Schritt V: Das Berechnungsmodell	108
6.8.1	Simulationen mit dem LB-Verfahren	108
6.9	Schritt VI: Die Visualisierung.....	109
6.10	Schlussfolgerung und Ausblick	111
Kapitel 7	Simulationsbeispiele	112
7.1	Klimatechnische Auslegung eines Gebäudes	112
7.2	Klimaanlagensteuerung über MSR-Technik.....	117
7.3	Luftströmung innerhalb einer Flugzeugkabine	119
Kapitel 8	Schlussfolgerung und Ausblick	121
8.1	Schlussfolgerung	121
8.2	Ausblick	124
Appendix A	Hardware	125
Literatur		129

Kapitel 1

Einführung

1.1 Problemstellung

Planung, Entwurf und Konstruktion eines (in der Regel nicht-standardisierten) Bauwerks im konstruktiven Ingenieurbau erfordert eine enge Kooperation zwischen den beteiligten Fachingenieuren und Architekten, um die Entwicklungs- und Planungsphase zeit- und kostenoptimal zu realisieren. Hohe Folgekosten früher Planungs-, Entwurfs- und Konstruktionsfehler¹ rechtfertigen Investitionen in eine frühzeitige und effektive Kommunikationsstruktur zwischen allen an diesem Prozess beteiligten Fachleuten [38]. Bislang ist für die Bauwirtschaft mehrheitlich festzustellen, dass durch unzureichende Kommunikation ein zielführendes Zusammenwirken sowie nachhaltige Qualitätskontrollen bei Planung und Entwurf erschwert werden.

Eine Planung im Bauwesen wird derzeit bauplanorientiert durchgeführt. Teilmodelle der Fachplaner werden als Zeichnung oder zweidimensionale CAD-Zeichnung erstellt. Bei fachübergreifenden Vorhaben wie beispielsweise computergestützten Berechnungen müssen Parameter und Attribute aus den Plänen der einzelnen Fachdomänen extrahiert und übertragen werden. Überdies potenziert bei einem iterativen Planungsprozess die fehlende Datenkonsistenz den negativen Effekt.

Der Grund hierfür liegt im Fehlen einer einheitlichen Informations- und Datenbasis der einzelnen Teilmodelle der Fachplaner. Eine Kompatibilität ist aufgrund ihrer unterschiedlichen Abstraktionsgrade und Divergenzen in Semantik, Definition und Strukturen nicht gewährleistet. Daraus folgt, dass eine isolierte informationstechnische Umsetzung der jeweiligen Fachmodelle in spezifische Softwarepakete zur Analyse und Prognose die Möglichkeit für eine integrative und damit effiziente Kommunikation ursächlich nicht zulässt.

Die logische Schlussfolgerung ist die Einführung eines allgemeinen *Datenmodells*, welches über einen hohen Abstraktionsgrad verfügt um die notwendigen Informationen aller Fachmodelle zusammen zu führen. Hiermit kann die Anzahl der Schnittstellen zwischen den beteiligten Fachplanern mit einer benötigten Mächtigkeit von $m \times n$ drastisch auf eine Schnittstellenanzahl von $m + n$ reduziert werden (siehe Abbildung 1.1).

¹ Laut Bauschadensbericht des Bundesministeriums für Raumordnung, Bauwesen und Städtebau sind 80 Prozent der Fehler im Bauwesen vermeidbar. Die Kosten für vermeidbare Bauschäden im Jahr 1992 liegen bei ca. 1,7 Mrd. € (als Verursacher sind Planer, ausführende Firmen und Baustoffhersteller aufgeführt). Ebenfalls lassen sich über die Hälfte aller Arbeitsunfälle am Bau laut Magazin BAULEX auf Planungsfehler und mangelnde Organisation zurückführen.

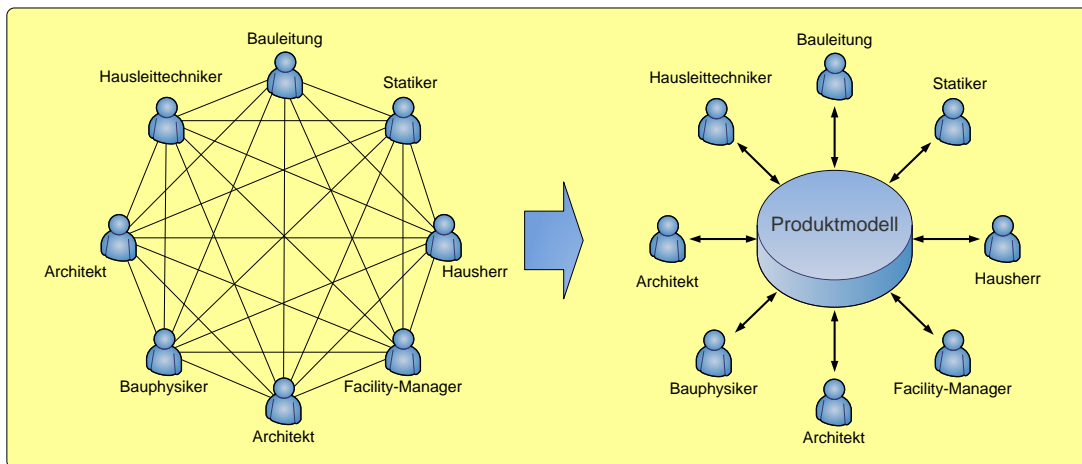


Abbildung 1.1: Optimaler Datenaustausch zwischen den am Planungsprozess beteiligten Fachplanern durch ein Produktmodell

Diese Idee verfolgt seit einigen Jahren im internationalen Bauwesen die IAI (INTERNATIONAL ALLIANCE FOR INTEROPERABILITY) [95] mit der Entwicklung der unter der ISO 16739 registrierten *Industry Foundation Classes (IFC)* [97]. Das *IFC Produktdatenmodell* dient hierbei als plattform- und applikationsunabhängiges Basismodell für den Austausch und die gemeinsame Nutzung von Daten. Es ist zu erwarten, dass sich die IFC als Standard zur digitalen Beschreibung von Gebäudemodellen etablieren werden.

Auch in Deutschland herrscht zunehmend Konsens, dass die IFC eine grundsätzlich geeignete, wenn auch unvollständige Basis für die informationstechnische Repräsentation kooperativer Planungs- und Konstruktionsaufgaben im Bauwesen darstellen. Mittlerweile bieten die Marktführer für CAD-Software, wie beispielsweise NEMETSCHEK, GRAPHISOFT, TEKLA, BENTLEY und AUTODESK entsprechende Schnittstellen für ihre Produkte an [93].

In einem weiterführenden Schritt kann dieses Produktdatenmodell genutzt werden, um eine Umgebung zur Optimierung der Planungsprozesse und Kommunikationswege zwischen den Fachplanern zu schaffen. Die Nutzung für den Austausch von Daten zwischen CAD-Programmen kann mit Methoden zur konsistenten Verwaltung von Produktdaten für verteilte, kooperative Projektabläufe erweitert werden. Diese stellen einen langjährigen Forschungsschwerpunkt der Bauinformatik dar [183, 215]. Die Umsetzung erfolgt unter anderem durch die Kopplung computergestützter Konstruktionswerkzeuge (Modellierer) und Programme zur *numerischen Simulation*. Es gibt Bemühungen, das IFC Modell als Grundlage für Simulationen von Raumluftrömungen zu nutzen [205], sowie weiterführend Ergebnisdaten von numerischen Simulationen aus dem Bereich der Strukturanalyse in das IFC Datenmodell zu integrieren [189, 194]. Durch die Nutzung dieser Simulationsprogramme ist eine Optimierung von Raumluftrömungen, wie sie in der Abbildung 1.2 dargestellt wird, unter Berücksichtigung energetischer und klimakomfortabhängiger Aspekte möglich.

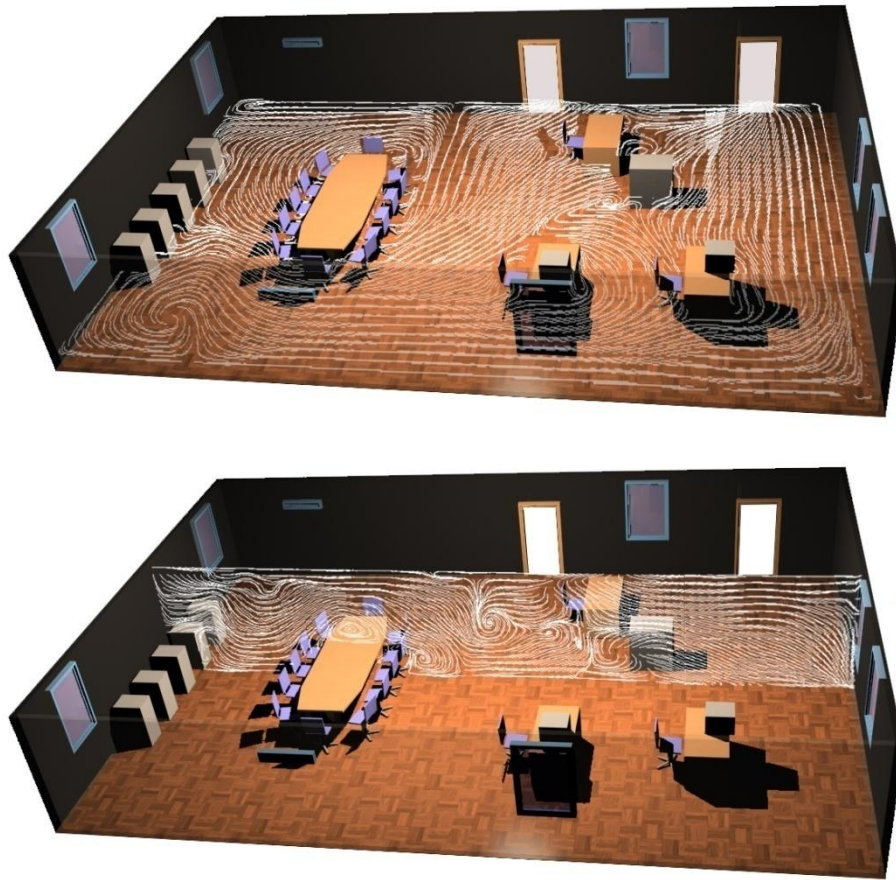


Abbildung 1.2: Die numerische Simulation: Berechnung von Raumluftströmungen

Die Optimierung von Raumluftströmungen wird aufgrund fehlender Beherrschbarkeit der Komplexität der physikalischen Zusammenhänge in gewöhnlichen Planungsprozessen nicht durchgeführt. Eine Schwierigkeit stellt hierbei der notwendige Modelltransfer dar. Aus dem vorhandenen Produktmodell muss für den entsprechenden Simulationskern ein Berechnungsmodell abgeleitet werden, welches dieser interpretieren kann. Für eine Strömungssimulation müssen ein volumenorientiertes Geometriemodell mit zusätzlichen Informationen zum Rechengebiet und bauteilspezifische und globale Randbedingungen für die Simulation vorliegen. Es ist erforderlich, dass das Bauwerksmodell diese Informationen besitzt, während das Modellierungswerkzeug die Daten extrahieren, analysieren und interpretieren kann. Dies bezieht sich sowohl auf die geometrische Struktur des Gebäudes als auch auf die einzelnen Bauteile. Das IFC Produktmodell hält über seine Struktur nicht die notwendigen standardisierten Attribute für die numerische Simulation vor. Über den objektorientierten und erweiterungsfähigen Ansatz des Modells wird allerdings die Voraussetzung zur Erweiterung durch eigene Attributdeklarationen geschaffen.

Für die Entwicklung eines *virtuellen Entwurfsraums*, bei dem die kooperative Zusammenarbeit zwischen den einzelnen Fachplanern im Vordergrund steht, soll der Konstruktions- sowie Simulationsprozess parallel ablaufen. Somit muss der beschriebene

Modelltransfer vollständig automatisiert ablaufen, um manuelles Einstellen von Parametern für Simulationsrandbedingungen zu vermeiden. Die Fachplaner können abhängig von der Zielsetzung gemeinsam über ihr jeweiliges, in einem globalen Modell zusammengeführtes Teilmodell einen Optimierungsprozess für ein Bauvorhaben durchführen.

Speziell die numerische Strömungssimulation bietet über die Kopplung mit zusätzlichen computergestützten Werkzeugen das Potenzial, bei der Lösung weiterer Problemstellungen innerhalb des Bereiches Bauwesen von Nutzen zu sein. Fachplaner können für eine optimale Klimaanlageauslegung unter Berücksichtigung von Energiekosten und dem thermischen Behaglichkeitsempfinden der Gebäudenutzer eine effiziente Unterstützung über die präsentierte Software erfahren. Hierbei muss für eine realistische Prognose über die Leistungsfähigkeit einer Klimaanlageinstallation neben der geometrischen Positionierung der Klimaanlagekomponenten im Raum auch deren Verhalten simuliert werden. Die Emulation der Verhaltensweisen solch einer regeltechnischen Anlage setzt eine interaktive Kopplung verschiedener Simulationsprogramme und die Nutzung innovativer Technologien voraus.

1.2 Zielsetzung der Arbeit

Im Rahmen dieser Arbeit wird der Ansatz verfolgt, durch kooperative und interaktive Simulation von Entwurfs- und Konstruktionsvarianten der einzelnen Fachmodelle, mit gleichzeitiger Visualisierung ihrer Zielfunktionswerte in *einem gemeinsamen virtuellen Entwurfsraum* eine gegenseitige Verständigung mittelbar zu ermöglichen und zu verbessern (siehe Abbildung 1.3).

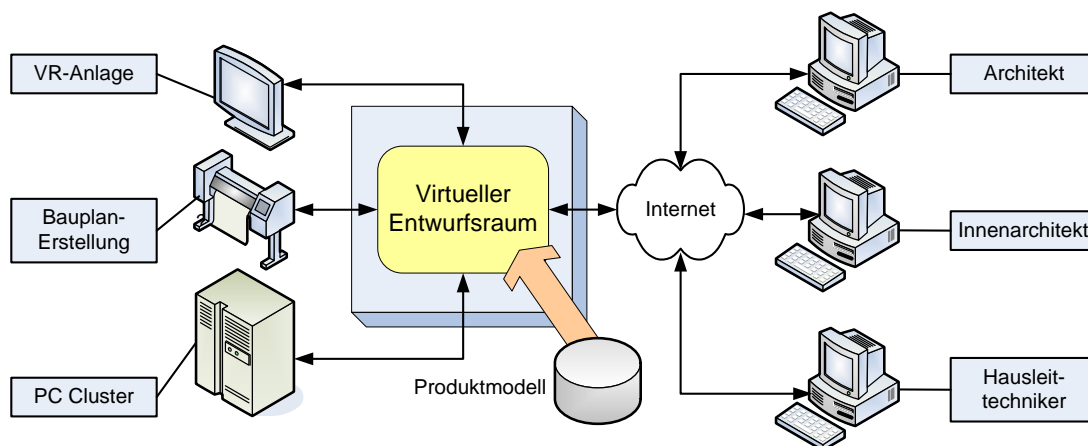


Abbildung 1.3: Vernetzt-kooperativer Planungsprozess

Durch schrittweise Modifizierung der einzelnen Fachmodelle (exemplarisch für den Architekt, Innenarchitekt und Hausleittechniker) ist eine interaktive Optimierung des Modells möglich. Die einzelnen Fachmodelle werden dabei in einem Produktmodell vorgehalten. Partielle Verbindungen und Abhängigkeiten zwischen den Objekten der

einzelnen Fachmodelle finden in entsprechenden Objekt- bzw. Klassendefinitionen ihren Ausdruck. Über eine gekoppelte Multi-User-Umgebung kann ein *vernetzter-kooperativer Zugriff* der Fachplaner ortsunabhängig auf den virtuellen Entwurfsraum erfolgen.

Solche Beziehungen werden im Allgemeinen nicht vollständig und eindeutig definiert, jedoch erlaubt selbst eine partielle Definition von Beziehungen zwischen Entitäten unterschiedlicher Fachmodelle die Ableitung algorithmisierbarer Regelwerke, die den Konstruktionsprozess durch Automatismen unterstützen.

Als Beispiel hierfür seien die Anschauungsobjekte Tür und Wand genannt: Aus statischer Sicht ist das Objekt Tür irrelevant; das Objekt Wand hingegen wäre genauer zu definieren als Scheibe, Schale oder beispielsweise elastisches Volumen. Diese aus statischer Sicht qualitative Eigenschaft hängt jedoch von geometrisch quantitativen Merkmalen des Objektes Wand ab. Demgegenüber stehen Beziehungen zwischen Tür und Wand, die einfach geprüft werden können: der standardisierte Öffnungsbereich einer Tür sollte beispielsweise schnittmengenfrei mit allen Wänden sein. Eine solche Sammlung von geforderten oder verbotenen Beziehungen wird schrittweise implementiert und resultiert in einem den Konstruktionsprozess unterstützenden, ‚intelligenten‘, offenen Expertensystem. Da alle wesentlichen Konstruktionsbestandteile geometrische Eigenschaften besitzen, eignet sich als Kern ein CAD-System. Dieses muss über eine Programmierschnittstelle erweiterbar sein und den Datenaustausch mit einem Produktmodell unterstützen.

Schwerpunkt ist hierbei die transparente Darstellung von Planungsentscheidungen der einzelnen Fachplaner im Gesamtzusammenhang. So können Abhängigkeiten in den verschiedenen Fachmodellen auch für spätere Phasen der Planung erkannt und negative wie positive Konsequenzen frühzeitig ermittelt werden.

Interaktive Elemente des Entwurfsraums wurden unter Verwendung des *Computational Steering* Ansatzes unter Einbezug einer *Virtual Reality*² Umgebung und eines *PC-Clusters* als Großrechenanlage realisiert (siehe hierzu Kapitel 5, Appendix A und [71, 202]). Computational Steering stellt Mechanismen zur Integration von Simulation, Datenanalyse, Visualisierung und Nachbearbeitung zur Verfügung. Sie gewährt dem Benutzer die interaktive Kontrolle über ein laufendes Programm. Ihr Operationsgebiet sind Anwendungen mit langen Laufzeiten, komplexe Simulationen, Datenmodellierungen und die Steuerung von Programmen in verteilten und parallelen Umgebungen [77, 119]. Dies bedeutet, dass die beteiligten Fachplaner durch direkte Modifikation eines Simulationssystems zur Laufzeit, ohne die Simulation zu verlassen, ein schnelles und tieferes Wissen der Auswirkungen dieser Änderungen erlangen können. Dies kann beinhalten, dass die zur Analyse eines Datensatzes eingesetzten Techniken interaktiv genutzt werden können [85]. Die Modifikation von Simulationsrandbedingungen oder der Geometrie des Simulationsgebietes, über welches das Gesamtsystemverhalten nach kurzer Antwortzeit zur Verfügung steht, ist ebenfalls möglich (Siehe hierzu Abschnitt 5.2 und [118, 119]).

² Als Virtuelle Realität oder engl. Virtual Reality (VR) wird die Darstellung und gleichzeitige Wahrnehmung der Wirklichkeit und ihrer physikalischen Eigenschaften in einer in Echtzeit computergenerierten virtuellen Umgebung bezeichnet.

Erst durch die Integration des Computational Steering Ansatzes lässt sich eine praxisnahe Konzeption und Optimierung von *HVAC-Systemen*³ [13, 177] realisieren. Hierbei werden *mess-, steuer- und regeltechnische Prozesse (MSR-Technik)* der Klimaanlageinstallation emuliert [56, 58, 61, 69, 72] und mit den virtuellen geometrischen Repräsentanten innerhalb des Entwurfsraums gekoppelt. Ausgehend von sensorisch zu erfassenden Größen wie Raumtemperatur, Luftfeuchtigkeit oder Luftgeschwindigkeit werden Flussstärke, Temperatur und Einfallswinkel der einströmenden Luft aus den Klimamodulen entsprechend angepasst. Diese dynamisch variierenden Zustände innerhalb eines Simulationslaufs, welche zwangsweise eine Rückkopplung zwischen dem Strömungssimulator und dem Klimaanlageinsimulator erfordern, sollen in weiten Teilen in den Optimierungsprozess einbezogen werden. Dies erfolgt über einen agentenbasierten Ansatz (siehe hierzu [69, 72] und Kapitel 4). Somit können, neben einer durchdachten Positionierung und Bemessung der Bauteile durch den Hausleittechniker, Regelkreise und Regelsteuerungen interaktiv innerhalb des Prototyps rein virtuell parametrisiert und optimiert werden.

1.3 Aufbau des Software Prototyps

Der Software Prototyp unterteilt sich in Module zur Modellierung, Berechnung, Visualisierung und HVAC Emulation sowie in notwendige Kopplungsschnittstellen für einen Datenaustausch zwischen den Modulen zur Modellierung und Berechnung und zwischen Berechnung und Visualisierung. Für die Kommunikation wurden jeweils unterschiedliche Techniken in Abhängigkeit von den jeweiligen Anforderungen gewählt. Die Abbildung 1.4 beschreibt den Aufbau des Prototyps.

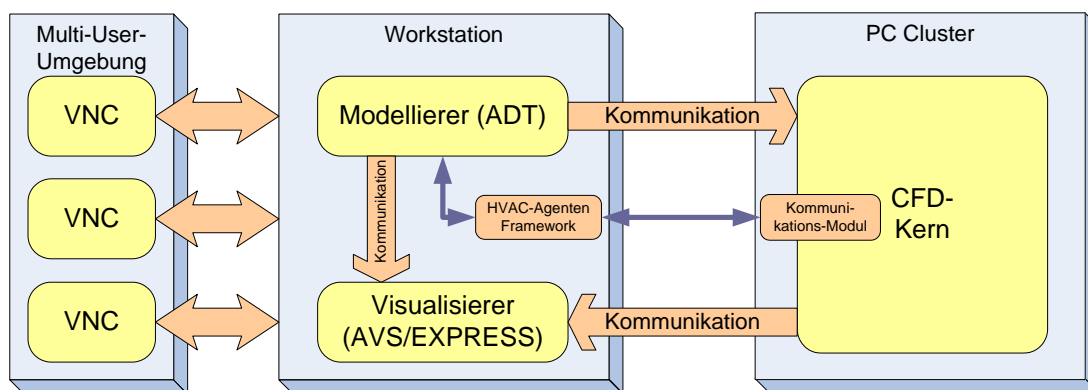


Abbildung 1.4: Systemaufbau des Software Prototypen

Das *Architectural Desktop (ADT)* [16] der Firma AUTODESK stellt die Basis für den virtuellen Entwurfsraum des Prototyps dar. Diese CAD-Anwendung wurde im Rahmen dieser Arbeit um eine Vielzahl notwendiger Funktionen erweitert. Dies umfasst die Kommunikationsschnittstelle zum Transfer der Simulationsdaten an den Rechenkern und

³ HVAC (engl.) ist die Abkürzung für „Heating, Ventilation and Air Conditioning“ und steht im Allgemeinen für den Begriff der Klimaregelung in Gebäuden. In einigen Regionen kann HVAC auch Leistungen von Rohrleitungssystemen und elektrischen Systemen beinhalten.

Überwachungswerkzeuge zur Kontrolle der Eingabeparameter für die Simulation. Dazu gehören auch geometrische Modifikationen während des Konstruktionsprozesses, Teile des Pre-Processing und die Integration eines Produktmodells als Datenbasis.

Eine weitere Kopplung an das ADT findet sich über das *HVAC-Agenten Framework*. Hierbei handelt es sich um eine eigenständige Softwareapplikation mit direktem Zugriff auf Objekte des virtuellen Entwurfsraums. Innerhalb dieser Applikation agieren autonome Teilsysteme, so genannte *Software-Agenten*⁴ (siehe [72, 217] und Kapitel 4), welche in einem Kooperationsverbund Aufgaben der *HVAC Mess-, Regel- und Steuertechnik* ausführen. Jeder Agent repräsentiert über die Konstruktionszeichnung ein Element der Klimaanlageinstallation. Die Abbildung von Systemen der Messsensorik und Raumluftregeltechnik ist hierüber möglich. Da dieser dynamische Simulationsvorgang eine permanente Rückkopplung der Agenten an den Rechenkern voraussetzt, ist das Framework über eine Kommunikationsschnittstelle an den Rechenkern angebunden.

Das Programm AVS/EXPRESS [6, 212] dient zur Visualisierung der Ergebnisdaten aus der Simulation und dem *Post-Processing*. Es handelt sich um ein Programm zur effizienten Darstellung großer Datenmengen. AVS besitzt eine leistungsfähige Programmierschnittstelle zur Erweiterung eigener Module. Die umfangreiche Modulbibliothek bietet *Isoflächen*, *Isovolumen* und *Stromlinien* zur Visualisierung von Luftströmungen.

Zwischen der Kommunikation von Simulationskern und der Visualisierungsumgebung sind aufgrund der großen numerischen Ergebnisdaten eine hohe Bandbreite sowie eine effiziente Kommunikation notwendig. Hierfür wurde das vom FORSCHUNGSZENTRUM JÜLICH entwickelte Programm VISIT [65, 66] herangezogen. Dieses „*Visualization Interface Toolkit*“ wurde speziell für den Transfer von Ergebnisdaten numerischer Simulationen innerhalb einer Computational Steering Umgebung entwickelt.

Bei dem in den Prototyp integrierten *CFD-Rechenkern*⁵ zur dreidimensionalen Simulation der Strömungs- und Temperaturfelder handelt es sich um eine adaptierte Eigenentwicklung der Software *Virtual Fluids* [127]. Diese basiert auf dem noch neuen *Lattice-Boltzmann (LB)*-Ansatz welcher primär für die Simulation von Mehrphasen- und Mehrkomponentensystemen in komplexer Geometrie entwickelt wurde [51, 52, 121-124, 186, 200, 203, 206, 207].

Für die effiziente Verarbeitung geometrischer Daten wird ein auf einem *Oktalbaum* [114] basierender Gittergenerator als *Pre-Processor* eingesetzt [219]. Aufgrund der hohen Anforderungen der Oktalbaumgenerierung an Rechenleistung und Speicherauslastung wurde dieser Prozess auf den leistungsstarken PC-Cluster übertragen.

Die Integration der Software VNC [175] als *Multi-User-Umgebung* erlaubt den kooperativen und verteilten Zugriff von verschiedenen Fachplanern auf ein gemeinsames Display. Der virtuelle Konstruktionsraum bzw. die Visualisierung wird hierbei als Fensterapplikation an beliebig viele Client-Rechner übertragen.

⁴ Die (Software-)Agenten-Technologie ist ein aus der Künstlichen Intelligenz (KI) entstandenes Konzept der Informatik.

⁵ CFD (engl. Computational Fluid Dynamics) = numerische Strömungsmechanik

1.4 Gliederung der Arbeit

Das Kapitel 2 beschreibt eine Einführung in die *Produktmodelle* und deren Struktur. Die Erläuterungen der AEC- und IFC- Produktmodelle, welche als Datenbasis des Software-Prototyps dienen, vermitteln damit das benötigte Vorwissen für die Einführung in die Funktionsweisen des *virtuellen Entwurfsraums* (Kapitel 3) und den *automatisierten Modelltransfer* (Kapitel 6).

Der kooperative Konstruktionsprozess wird über den in Kapitel 3 erläuterten *virtuellen Entwurfsraum* durchgeführt. Nach einer Einführung in den Aufbau der Programmierschnittstellen des als Modellierer eingesetzten AUTODESK Architectural Desktops (ADT) werden implementierte Funktionen zur Automatisierung und Kommunikation beschrieben.

Die Emulation mess-, steuer-, und regelungstechnischer Anlagen für Klimaanlagekomponenten erfolgt über das in Kapitel 4 beschriebene *HVAC-Agenten Framework*. Nach einer Einführung in die Themen der thermischen Behaglichkeit, der Regelungstechnik und der Software-Agenten-Technologie wird die Umsetzung und Funktionsweise der HVAC-Agenten im Verbund mit dem Software-Prototyp erläutert.

Das Kapitel 5 beschreibt die Techniken, welche zur Umsetzung kooperativer und interaktiver Prozesse eingesetzt wurden. Hierbei werden Prinzipien und Funktionsweisen von Virtual Reality Anlagen, Computational Steering Systemen und Multi-User-Umgebungen beschrieben.

Zur Ableitung eines IFC Produktmodells in ein Berechnungsmodell für die numerische Simulation sind eine Vielzahl von Zwischenschritten notwendig. Der in Kapitel 6 beschriebene *Modelltransfer* erläutert den sequenziellen Ablauf dieser automatisierten Teilschritte. Zum besseren Verständnis der Konvertierungsprozesse geometrischer Modelle beginnt das Kapitel mit einer Einführung in die dreidimensionale geometrische Modellierung.

Die Ergebnisse einer Anzahl über den Software-Prototyp durchgeführter Simulationen werden in Kapitel 7 aufgeführt. Das Kapitel 8 schließt die Arbeit mit einer Zusammenfassung und einem Ausblick ab.

Kapitel 2

Produktmodelle

2.1 Einleitung

Zu Beginn einer Bauwerksplanung werden im Allgemeinen von den beteiligten Fachplanern spezifische Modelle erstellt, welche nur eine bestimmte Sicht auf das Bauwerk zulassen. Ein Architekt oder Statiker entwickelt sein Bauwerksmodell unter geometrischen Gesichtspunkten. Ein Hausleittechniker benötigt nur eine symbolische Darstellung von Schaltungen und Anlageninstallationen und ist nicht auf einen geometrisch korrekten Grundriss angewiesen. Unter baubetrieblichen Gesichtspunkten wird ein Modell als Betriebsprozess beispielsweise in Form einer Bauablaufplanung dargestellt. In der Praxis ist der Austausch dieser Modelle zwischen den am Projekt beteiligten Unternehmen ein notwendiger Schritt zur effizienten Pflege der Baupläne und der darin enthaltenen Zusatzinformationen. Dies geschieht im Allgemeinen mit unterschiedlichen und teilweise inkompatiblen CAD-Programmen. Dieses Vorgehen führt zwangsweise zu einer redundanten Datenhaltung und im Falle von Planungsänderungen häufig zu Informationsinkonsistenzen.

Eine evolutive Entwicklung im Bauwesen ist die Einführung von Metainformationen für Bauteile über ein Datenmodell, welches sämtlichen fachspezifischen Partialmodellen übergeordnet ist und diese zusammenfasst. Dieses schnittstellenstandardisierte und plattformunabhängige Gesamtmodell kann über eine spezifische Unternehmenssoftware (CAD- und nicht-CAD Programme) eingelesen und verarbeitet werden. Hierbei beinhaltet dieses Modell, ein in der Abbildung 2.1 dargestelltes *Produktmodell*, nicht nur Informationen zur Herstellung des Gebäudes, sondern auch relevante Daten zur Wartung und Handhabung von Gebäudeteilen (*Facility-Management*) bis hin zur Entsorgung.

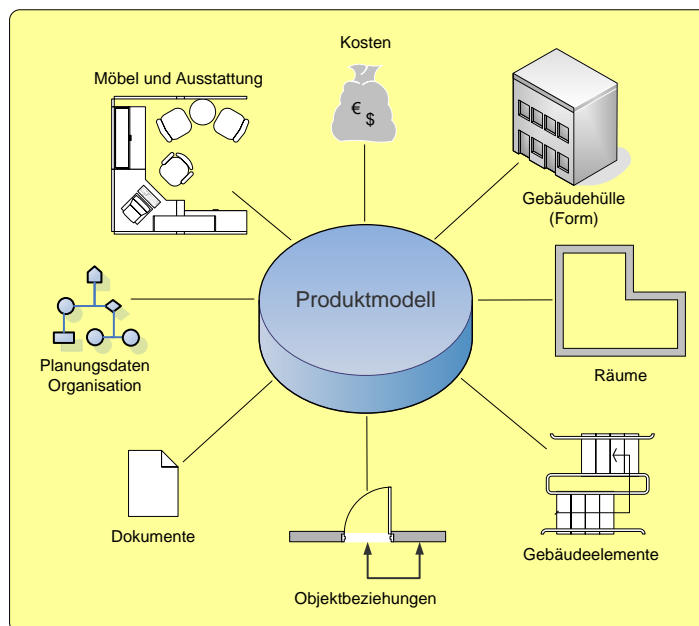


Abbildung 2.1: Konzept eines Produktmodells; Integration planungsrelevanter Metainformationen

Inkonsistenzen können über den konzeptionellen Ansatz des Modells (objektorientiert; nicht-redundante Datenhaltung) ausgeschlossen werden. Durch die Einführung bauteilspezifischer Metainformationen erhalten die Objekte in einem Produktmodell einen gewissen Grad an „Intelligenz“. Das zu Grunde liegende Prinzip wird über die Abbildung 2.2 erläutert.

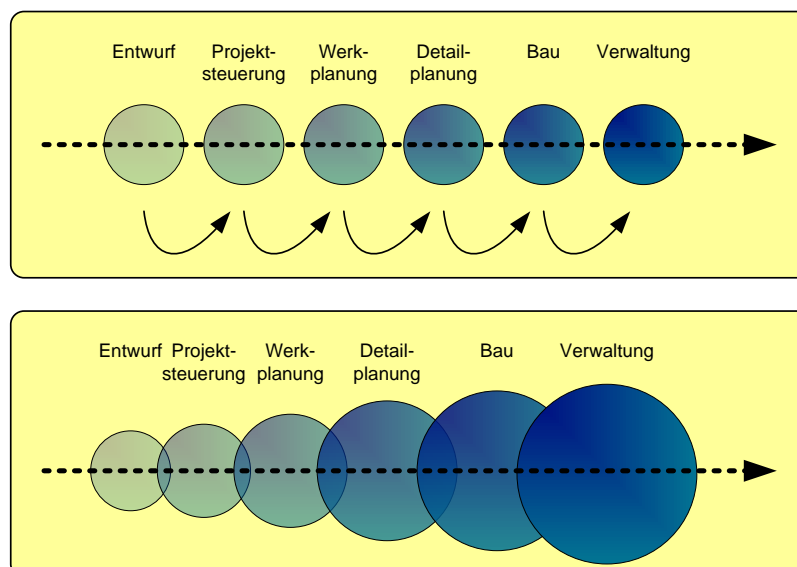


Abbildung 2.2: Daten werden intelligent; Früher (oberer Teil): Redundante Daten, doppelte Arbeit, Fehler durch falsche Kommunikation; Heute (unterer Teil): Datenintelligenz wächst, jeder Projektbeteiligte ergänzt das Datenmodell

Beispielsweise wird eine Wand nicht wie in einer traditionellen CAD-Zeichnung über ein geometrisches Objekt aus einer Ansammlung von Punkten und Linien definiert. Sie erhält zusätzliche semantische Informationen, um mit anderen Objekten interagieren zu können, oder gibt Aufschluss über ihre Zusammensetzung. Die Wand *kennt* beispielsweise die in ihr verbauten Fenster und Türen. Wird ein Fenster vergrößert, passt das Objekt Wand automatisch die Rohbauöffnung für das Fenster an. Die Wand kann sich durch den Zusammenschluss benachbarter Wände einem Gebäuderaum zuordnen. Analog ordnen sich die Objekte Fenster und Tür dem Objekt Wand zu und bewegen sich bei einer Verschiebung der Wand mit. Durch das Anhängen zusätzlicher Objektattribute besitzt die Wand Informationen über ihr Mauerwerksmaterial, ihre Fertigungskosten, ihre statischen Eigenschaften oder ihr energetisches Verhalten. Hierbei wird auch die Geometrie des Bauteils als Attribut eines Objektes verstanden. Ein Objekt kann mehrere geometrische Formen besitzen. Abhängig von spezifischen Anforderungen an das Aussehen eines Objektes kann die dafür benötigte Geometrie gewählt werden. Beispielsweise kann das Objekt Wand als dreidimensionaler Volumenkörper für einen „virtuellen Rundgang“ durch das Gebäude dargestellt werden. Zusätzlich ist eine zweidimensionale Darstellung des Objektes für die Erstellung eines Bauplans über ein weiteres geometrisches Attribut möglich.

Produktmodelle und Produktdatenmodelle. Generell wird zwischen dem Begriff „Produktmodell“ und „Produktdatenmodell“ unterschieden. Nach Definition der Informatik stellt ein Produktdatenmodell hierbei das Datenschema als Grundgerüst zur Beschreibung von Datenstrukturen, Eigenschaften und Beziehung einzelner Objekte zueinander bereit. Die Definition des Aufbaus der Elemente ergibt in ihrer Gesamtheit das „Produkt“ eines Gesamtbauwerks. Im Sinne der *objektorientierten Programmierung* kann also bei einem Produktdatenmodell von einer *Klasse* gesprochen werden, aus deren Instanziierung sich das *Objekt* Produktmodell ergibt.

Die Struktur von Produktdatenmodellen bilden hierarchische *Schichten (Layer)*, bei denen das Gesamtdatenmodell in logische Klassen, die *Partialmodelle (Domains)*, unterteilt wird. Ein Partialmodell beschreibt über seine Datenstruktur einen Teilaspekt des Produktes und kann über definierte Schnittstellen einen Bezug auf andere Partialmodelle aufnehmen. Dieser Aufbau ermöglicht eine Erweiterung des Modells. Im Maschinenbau sowie in der Automobilindustrie gilt *STEP*⁶ [104, 105, 107] als Standard für ein Produktdatenmodell. Im Bereich des Bauwesens ist die schrittweise Etablierung der *Industry Foundation Classes (IFC)* [97] der IAI [95] als Standard zu beobachten.

⁶ Standard for the Exchange of Product Model Data (STEP)

2.2 Das STEP Produktmodell

Bei dem über die ISO 10303 dokumentierten STEP handelt es sich um ein komplexes Normenwerk⁷ zur Beschreibung grundlegender Methoden der Produktmodellierung sowie notwendiger Anwendungswerkzeuge. Ziel der seit 1994 laufenden Entwicklung ist eine eindeutige, computerbasierte Darstellung von Produktinformationen über den gesamten Lebenszyklus eines Modells. Der Lebenszyklus ist in die Phasen Entwicklung, Fertigung, Vertrieb, Wartung und Entsorgung eingeteilt [9]. Über STEP findet die Idee des *integrierten Produktmodells* Anwendung und beschreibt die Zusammensetzung eines Produktmodells aus schnittstellenverknüpften Partialmodellen. Das als Datenaustauschformat geplante STEP sichert die Konsistenz und Redundanzfreiheit der Daten durch den zusätzlichen Austausch des Modellaufbaus (Modelldatenschema) neben den Produktinformationen.

Der Aufbau des STEP Normenwerks setzt sich neben der Produktmodelldefinition (Informationsmodellen) auch aus Modulen zu Spezifikations- und Implementierungsmethoden, Anwendungsprotokollen sowie Konformitätsprüfungsmethoden zusammen. Zur Umsetzung eines branchenspezifischen Produktmodells und anwendungsspezifischer Erweiterungen werden benötigte Teilinformationen extrahiert. Hierfür werden Applikationsprotokolle (AP) zur Verfügung gestellt, wie beispielsweise das AP 214 (Core Data For Automotive Mechanical Design Processes), das AP 218 (Ship Structures), das AP 233 (Systems Engineering Data Representation), das AP 236 (Furniture Product Data And Project Data) oder das für das Bauwesen relevante AP 225 (Building Elements Using Explicit Shape Representations). Die Applikationsprotokolle ermöglichen den Einsatz verschiedener Systeme wie CAD, CAM, PDM, DMU und CAE. Die nachfolgende Abbildung 2.3 verschafft einen Überblick über die wesentlichen Bestandteile des STEP Normenwerks.

⁷ Das Normenwerk STEP besteht aktuell aus einem über 2500 Seiten umfassenden Werk, welches in 1760 Teilspezifikationen (Parts) unterteilt ist.

ISO TC184 SC4

STEP on a Page

ISO 10303

APPLICATION PROTOCOLS AND ASSOCIATED ABSTRACT-TEST SUITES

I 201 Explicit draughting [ATS 301 = X] I 202 Associative draughting [X] I 203 Configuration-controlled design (c2=I,a1=I)[X] I 204 Mechanical design using boundary rep [I] X 205 Mechanical design using surface rep [W] X 206 Mechanical design using wireframe [X] I 207 Sheet metal die planning and design [I] X 208 Life-cycle product change process [X] I 209 Composite & metal structural anal & related design[X] I 210 Electronic assy, interconnection & packaging design [X] X 211 Electronic P-C assy: test, diag, & remanuf[X] I 212 Electrotechnical design and installation [C] X 213 Num control (NC) process plans for mach'd parts [X] I 214 Core data for automotive mech design processes (e2=E)[F] E 215 Ship arrangement [X] E 216 Ship moulded forms [X] X 217 Ship piping [X] E 218 Ship structures [X] X 219 Dimension inspection [X] O 220 Proc. plg, mfg, assy of layered electrical products [X]	C 221 Functional data & their schem rep for process plant [X] X 222 Design-manuf for composite structures [W] X 223 Exch of design & mfg product info for cast parts [@] I 224 Mech pdt def for p. plg using mach'n'g feat (e2=X,e3=A) I 225 Building elements using explicit shape rep [C] \[X,I] X 226 Ship mechanical systems [C] I 227 Plant spatial configuration(e2=C) [X] X 228 Building services: HVAC [X] X 229 Design & mfg product info for forged parts[X] X 230 Building structural frame: steelwork [X] X 231 Process-engineering data [X] I 232 Technical data packaging: core info & exch [I] W 233 Systems engineering data repr (to be PAS 20542)[X] X 234 Ship operational logs, records, and messages[X] W 235 Materials info for des and verif of products [X] W 236 Furniture product and project data[W] W 237 Computational Fluid Dynamics A 238 Computer numerical controllers W 239 Product life-cycle support W 240 Process plans for machined products
--	---

COMMON RESOURCES (with 13584-20 logic. model of expr.(I) and 15531-42 Time (W))

APPLICATION MODULES (Technical specifications)

For status of the modules access the file via the SOAP home page.

Legend: TS Status

0-10 =O=prop-->apvl for ballot
10-20=A=NP blt circ-->NP apvl
20-60=D=DTS dev-->reg as TS
>60 =T=TS Published

INTEGRATED-APPLICATION RESOURCES

I 101 Draughting (c1=I) X 102 Ship structures X 103 E/E connectivity I 104 Finite element analysis I 105 Kinematics (c1=I, c2=I)	X 106 Building core model C 107 Finite-element analysis definition relationships C 108 Prmetizat'n&Constraints for expl geom prod mdl C 109 Assembly model for products W 110 Mesh-based computational fluid dynamics
--	---

INTEGRATED-GENERIC RESOURCES

I 41 Fund of prdet descr & spt (e2=I,c1=I) I 42 Geom & top rep (c3=I,e2c1=I,e3=F) I 43 Repres specialization (e2=I,c1=I,c2=I) I 44 Product struct confg (e2=I,c1=I) I 45 Materials (c1=I) I 46 Visual presentation (c1=I, c2=I) I 47 Tolerances (c1=I) X 48 Form features I 49 Process structure & properties	I 50 Mathematical constructs E 51 Mathematical description W 52 Mesh-based topology W 53 Numerical Analysis C 54 Classification Set theory A 55 Procedural and hybrid represent. W 56 State W 57 Expression extensions A 58 Risk
---	--

APPLICATION-INTERPRETED CONSTRUCTS

I 501 Edge-based wireframe I 502 Shell-based wireframe I 503 Geom-bounded 2D wireframe I 504 Draughting annotation I 505 Drawing structure & admin. I 506 Draughting elements I 507 Geom-bounded surface I 508 Non-manifold surface I 509 Manifold surface I 510 Geom-bounded wireframe I 511 Topological-bounded surface	I 512 Faceted B-representation I 513 Elementary B-rep I 514 Advanced B-rep I 515 Constructive solid geometry X 516 Mechanical-design context I 517 Mech-design geom presentation(c1=I) I 518 Mech-design shaded presentation I 519 Geometric tolerances (c1=I) I 520 Assoc draughting elements @521 Manifold subsurfaces E 522 Machining features A 523 Curve swept solid
---	--

IMPLEMENTATION METHODS

I 21 Clear-text encoding exch str (c1=I,e2=I) I 22 Standard data access interface I 23 C++ language binding (to #22) I 24 C language binding (to #22)	C 25 EXPRESS to OMG XMI X 26 IDL language binding (to #22) I 27 JAVA language binding (to #22) @28 XML rep for EXPRESS-schemata & data X 29 Ltwt Java binding (to #22) \ (DTS)
--	--

DESCRIPTION METHODS

I 1 Overview and fundamental principles
I 11 EXPRESS language ref man. (c1=I,e2=C e3=X ISO 20303-X a1=X)
I 12 EXPRESS-I language ref man (Type 2 tech report, not a 10303 part)
X 13 Architecture and Methodology reference manual
I 14 EXPRESS X Language reference manual

CONFORMANCE TESTING METHODOLOGY & FRAMEWORK

I 31 General concepts
I 32 Requirements on testing labs and clients
X 33 Structure and use of abstract test suites
I 34 Abstract test methods for Part 21 implementation.
C 35 Abstract test methods for Part 22 implementation.

Legend: Part Status (E, F, I safe to implement)
0=O=Preliminary Stage (Proposal-->appr for NP ballot)
10=A=Proposal Stage (NP ballot circ-->NP approval)
20=W=Preparatory Stage (Wkg Draft devel.-->CD regis)
30=C=Committee Stage (CD circulation-->DIS regis)

40=E=Enquiry Stage (DIS circ.-->FDIS registration)
50=F=Approval Stage (FDIS circ-->Int'l Std regis)
@=At ISO, approved for publication (ISO status 40.95 or 50.99)
60=I=Publication Stage (Int'l Std published)
98=X=Project withdrawn

Abbildung 2.3: „STEP on a Page“ [157]; Übersicht über die Bestandteile des STEP Normenwerkes

Zur Modellierung von Datenstrukturen wird die in den Beschreibungsmethoden (10er Serie) spezifizierte Beschreibungssprache *EXPRESS* (ISO 10303-11, [107]) angewandt. Über ihren objektorientierten Ansatz ist die Abbildung von Klassen (*Schemas*) und Klassenbeziehungen (*Schema-Interoperability*), Objekten (*Entities*) mit entsprechenden Eigenschaften (*Attributes*), Vererbungsregeln (*Inheritance*) und Integritätsbedingungen (*Rules*) möglich. *EXPRESS* ist Bestandteil von STEP, findet aber auch in anderen Bereichen der Datenmodellierung Anwendung. Für einen Überblick über STEP sei auf die Arbeiten von NEUBERG und ANDERL verwiesen [9, 161].

Im Bereich des Bauwesens hat sich STEP nicht durchgesetzt, da sich die Hauptausrichtung der Normierungsbemühungen im Rahmen der ISO 10303 auf die Automobilindustrie sowie den Maschinenbau bezieht. Als Konsequenz wurden auf Seiten der STEP-Entwickler die bauspezifischen Applikationsprotokolle AP 228 (Building Services: HVAC Modelling), AP 230 (Building Structural Frame: Steelwork) sowie der Teil 106 (Building and Construction: Core Model) aus STEP eliminiert und auf Seiten des Bauwesens eine unabhängige Organisation zur Definition eines eigenen Produktmodells gegründet. Über diesen Schritt erfolgte 1995 die Initiierung der im nachfolgenden Abschnitt beschriebenen Industry Foundation Classes (IFC) der IAI.

2.3 Das IFC Produktmodell

Ein Grund für die geringe Beachtung von STEP im Bauwesen liegt unter anderem in den Zieldefinitionen bezogen auf den Austausch geometrischer Informationen. Hier werden Extrusionsmodelle⁸, welche in der Bauindustrie die häufigste Anwendung finden nach den Spezifikationen der STEP nicht unterstützt. Geometrische Objekte werden im STEP-Format über im Maschinenbau übliche NURBS⁹ beschrieben.

Mit dem Gedanken zur Definition eines Basismodells, welches einen Datenaustausch für die gemeinsame Datennutzung in der Bauindustrie und im Facility Management (FM) unterstützt, wurde die INTERNATIONAL ALLIANCE FOR INTEROPERABILITY (IAI) [95] gegründet. Die Mission der IAI ist die Schaffung einer Basis zur Verbesserung der Planungs- und Ausführungsprozesse durch eine gemeinsame Datenhaltung und einen Datenaustausch. Das Werkzeug hierfür stellen die nach ISO/PAS 16739 normierten IFC dar, welche allen Mitgliedern der IAI zugänglich sind und damit allen interessierten AEC¹⁰ Software-Unternehmen offen stehen. Über die IFC ist es möglich, den Produktdatenaus-

⁸ Die Extrusion einer Geometrie bezeichnet eine Dimensionserhöhung einer zweidimensionalen Form durch Parallelverschieben im Raum. Die Extrusion einer Fläche ergibt einen Körper mit dem Querschnitt der Fläche. Somit ergibt die Extrusion eines Quadrates einen Quader und die Extrusion eines Kreises einen Zylinder.

⁹ NURBS: Non-Uniform Rational B-Splines (Freiformkurven) sind mathematisch definierte Kurven oder Flächen zur Modellierung beliebiger Formen. Die Darstellung der Geometrieinformation erfolgt über stückweise funktional definierte Geometrieelemente. Über dieses Prinzip kann jede beliebige technisch herstellbare oder in der Natur vorkommende Form mit Hilfe von NURBS abgebildet werden.

¹⁰ AEC (engl. Architecture, Engineering and Construction) beschreibt alle computergestützten Anwendungen in Architektur, Bauingenieurwesen (Hoch- und Tiefbau), Energietechnik, Stahl- und Anlagenbau sowie Fabrik- und Büroplanung.

tausch zwischen allen am Bau beteiligten Fachplanern zu gewährleisten. Die in den vorangegangenen Phasen gesammelten Daten können in jeder Bauphase mit einbezogen werden.

Die IAI ist ein weltweit agierender Verbund von Firmen aus dem Gesamtumfeld der Planung, des Bauens und der Bewirtschaftung von Gebäuden. Die Mitglieder sind Planer, Fachingenieure, Bauherren, Bauunternehmen, Softwarehäuser, Verbände, Hochschulen und Forschungseinrichtungen sowie Betreiber und Nutzer von Gebäuden. Mehr als 600 Unternehmen arbeiten an der Optimierung der Industry Foundation Classes. Die IAI wurde im Juni 1995 in den USA gegründet. Kurz darauf folgten Gründungen in Deutschland, Großbritannien, Frankreich, Skandinavien, in Singapur, Japan, Korea und Australien. Die IAI (international) wird durch das INTERNATIONAL COUNCIL (IC) geleitet und über ihre Landesvertreter, welche in *Chapter* eingeteilt sind, repräsentiert. Gemeinsame technische Arbeiten der IAI werden über das INTERNATIONAL TECHNICAL MANAGEMENT COMMITTEE (ITM) koordiniert. In den deutschsprachigen Ländern der Bundesrepublik Deutschland, Österreich und der Schweiz, werden die Interessen der Firmen dieser Länder durch die INDUSTRIEALLIANZ FÜR INTEROPERABILITÄT E.V. [99] vertreten.

2.3.1 Aufbau und Struktur

Die Struktur des IFC-Modells basiert auf dem aus der ISO 10303 STEP übernommenen „Building and Construction Core Model“ welches unter Berücksichtigung der Erfahrungen aus den Teilprojekten ATLAS [14, 32, 171] und COMBI [47, 116, 181, 182] des EU-Projekts ESPRIT weiterentwickelt wurde. Somit erscheint aus historischer Sicht logisch, dass sich das IFC Produktmodell der abstrakten Beschreibungssprache EXPRESS bedient und über eine im ASCII-Standard kodierte Textdatei nach ISO 10303-21 [105] veröffentlicht wird (Abbildung 2.4). Diese Methoden stellen Ergebnisse der im vorangegangenen beschriebenen Projekte dar.

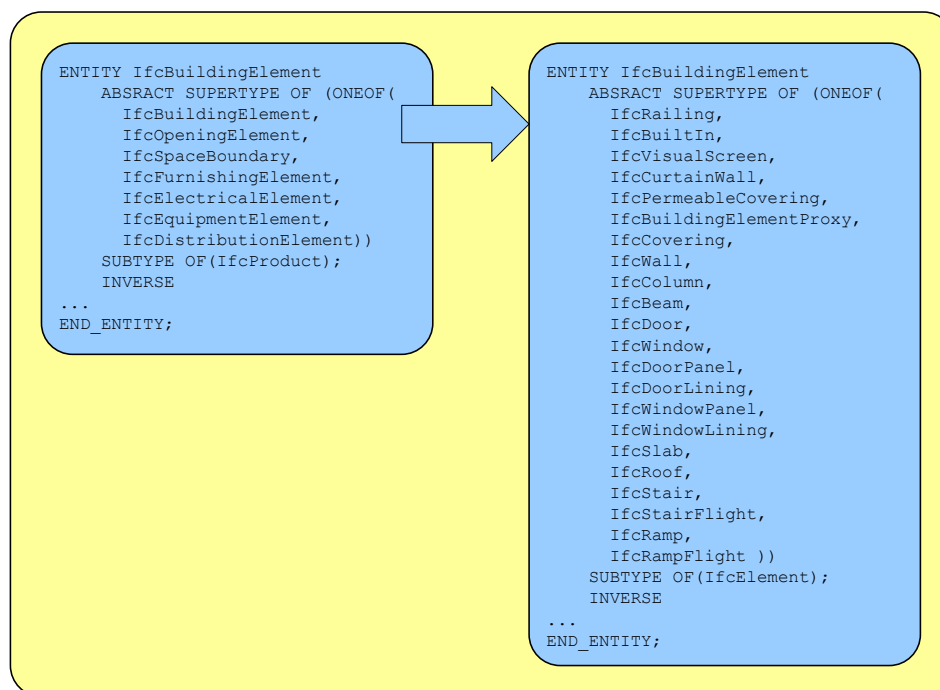


Abbildung 2.4: Darstellung nach EXPRESS Spezifikationen als ASCII-Standard kodierte Textdatei

Zur Darstellung der komplexen Produktmodelle wird die grafische Variante EXPRESS-G [96] genutzt, welche Konstrukte wie *Entities*, *Attribute* und *Typendeklarationen* sowie *Vererbungshierarchien* als Teilmenge von EXPRESS ermöglicht. Ihre in der Abbildung 2.5 dargestellte Struktur weist Analogien zu der schematischen Beschreibungsform über UML¹¹ [164] auf.

¹¹ Die Unified Modeling Language (UML) ist eine von der Object Management Group (OMG) entwickelte und standardisierte Sprache für die Modellierung von Software und anderen Systemen.

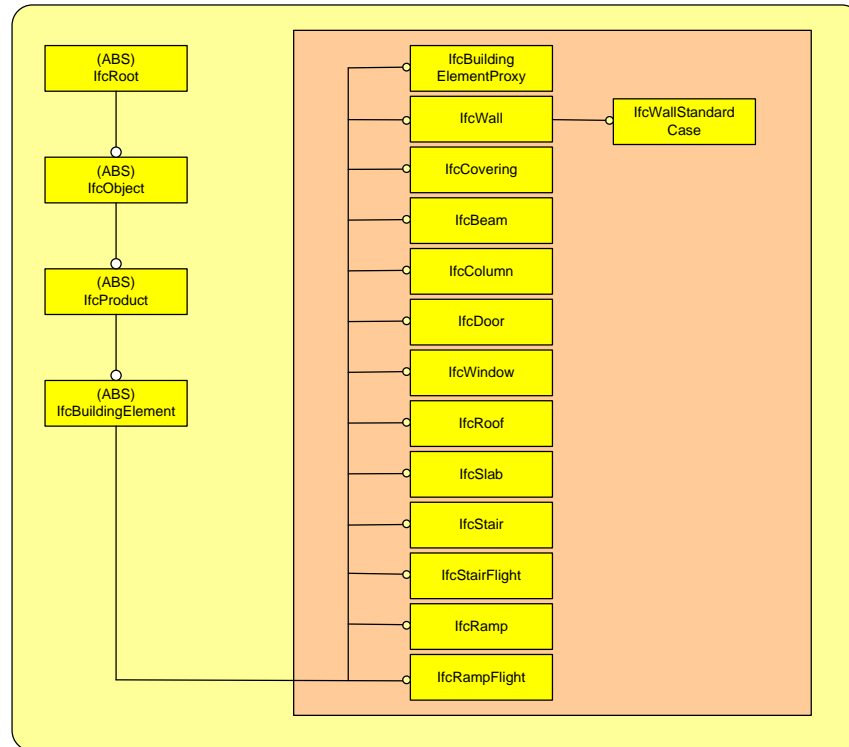


Abbildung 2.5: Darstellung nach EXPRESS-G Spezifikationen als Strukturgrafik

Bauteile (*Entities*), welche gleiche Eigenschaften aufweisen, werden zu Klassen zusammengefasst. Aus diesen Klassen instanziierte Objekte (beispielsweise eine Wand) werden IDs zugewiesen. Über die Zuteilung einer *GUID*¹² [167] ist die eindeutige Identifizierung über die gesamte Lebensdauer des Bauteils gewährleistet.

Die IFC sind über einen hierarchischen Ansatz aufgebaut, welcher die Struktur des Modells in vier Hauptschichten (*Layer*) aufteilt. Die Abbildung 2.6 stellt den Aufbau der aktuellen IFC 2x3 Spezifikationen aus dem Jahr 2006 dar.

¹² Die GUID (Globally Unique Identifier) auch unter dem Namen UUID (Universal Unique Identifier) kommuniziert und basiert auf einem Algorithmus, welcher eine global eindeutige Nummer mit Hilfe des aktuellen Datums, der Uhrzeit, einer Netzwerkkarten-ID und einem Wert eines Hochfrequenzzählers errechnet. Es besteht statistisch gesehen keine Möglichkeit, dass dieser Algorithmus bis in das Jahr 3400 eine doppelte ID erzeugt.

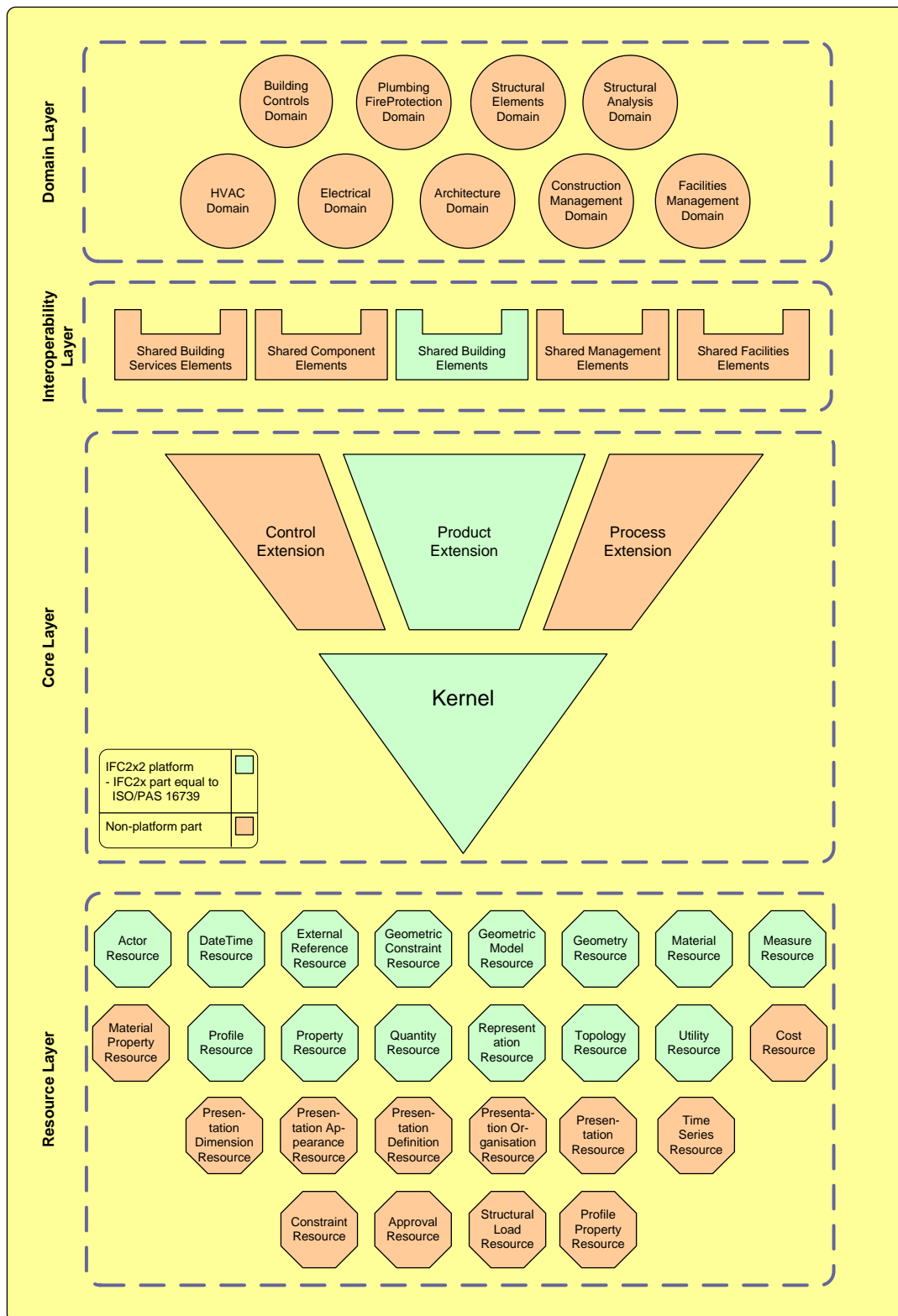


Abbildung 2.6: Aufbau der IFC 2x3 Spezifikation

Der Detaillierungsgrad des Modells nimmt von der unteren zur oberen Ebene zu:

- **Resource Layer:** Die unterste Schicht bilden die Ressourcen, welche sich aus allgemeingültigen Konzepten und Klassen zusammen setzen und keine Abhängigkeiten zu einem Anwendungsbereich (*Domain*) aufweisen. Sie können von anderen Klassen höherer Schichten für die Instanziierung eines Objektes herangezogen werden. Teilweise werden für die Erstellung zusätzliche Klassen benötigt (beispielsweise *Geometry Resource*).
- **Core Layer:** Der *Kernel* sowie dessen Erweiterungen *Control Extension*, *Product Extension* und *Process Extension* bilden die zweite Schicht des Modells. Grundlegende Strukturen und Konzepte des IFC-Modells werden hierüber definiert und in den nachfolgenden Schichten präzisiert.
- **Interoperability Layer:** Zur Gewährleistung der Interoperabilität zwischen den einzelnen Domänen stellt der Interoperability Layer Schemata und Klassen bereit, welche mindestens zwei unterschiedliche Anwendungsbereiche fordern. Die Schicht bildet die Schnittstelle zur Überführung anderer Anwendungsmodelle in das IFC-Modell. Dies erfolgt über definierte *Schema Maps* (Abbildungsvorschriften).
- **Domain Layer:** Die oberste Schicht des IFC-Modells bildet der Domain Layer zur Spezialisierung der Objekte für einen konkreten Anwendungsbereich.

Objekte einer Schicht referenzieren analog den Vererbungsregeln der objektorientierten Programmierung nur auf Objekte eines darunter liegenden Layers. Dies gewährleistet die Erweiterung des *Domain Layers* ohne eine notwendige Modifikation der Objekte des *Interoperability Layers* und *Core Layers*.

Die Klasse *IfcRoot* (Kernel) besitzt den höchsten Abstraktionsgrad des IFC-Modells. Sie ist über einen Namen, eine ID, eine Beschreibung und eine Historie definiert. Eine Subklasse von *IfcRoot* stellt *IfcObject* dar. Diese repräsentiert konkrete und konzeptionelle Objekte einer Domain wie beispielsweise eine Wand, einen Arbeitsplan, die Kosten einer Position, Laborergebnisdaten oder eine Person. Weitere Subklassen von *IfcRoot* sind *IfcPropertyDefinition* zur Darstellung der Objekteigenschaften und *IfcRelationship* zur Definition von Beziehungen zwischen Objekten des Typs *IfcObject* und der Beziehung zwischen Modellklassen. Die Kernel Subklasse *IfcObject* kann in der nächsten Ebene über die Kernel-Erweiterungen des *Core Extensions Layers* (innerhalb des *Core Layers*) spezialisiert werden. Dies geschieht über die Ableitung in eine der drei Subklassen *IfcControl*, *IfcProcess* und *IfcProduct*.

Als Beispiel sei erneut das Objekt „Wand“ genannt. Die geometrischen Informationen werden mit anderen *Entites* wie Stützen oder Platten in der Klasse *IfcProduct* zusammengefasst. Sie teilen gewisse Eigenschaften wie die Position im Raum (*IfcLocalPlacement*) oder die geometrische Darstellung (*IfcProductRepresentation*). Objekte der Klasse *IfcProduct* können bis zu einem gewissen Grad in Unterklassen aus dem *Resource Layer* unterteilt werden, so beispielsweise die Resource-Klasse *IfcCartesianPoint* für die Abbildung von Koordinaten eines kartesischen Punktes.

2.3.2 Entwicklung

Die Abbildung 2.7 stellt einen Überblick über die historische Entwicklung der Spezifikationen des IFC-Produktmodells dar.

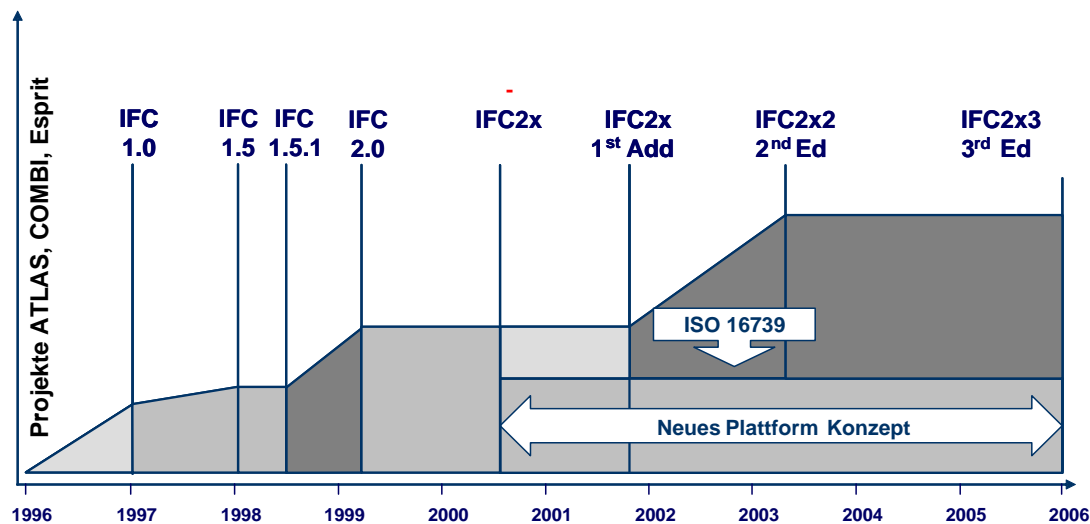


Abbildung 2.7: Historie und Entwicklungsstand der IFC Spezifikationen

Die Entwicklung der IFC begann im Jahre 1996. Die erste prototypische Version der IFC wurde 1997 veröffentlicht. Einen Meilenstein stellt das Release der Version 1.5.1 von 1998 dar, bei dem es sich um die erste kommerzielle Implementierung handelt deren Ziel das Austauschen von Rohbaudaten darstellt. Eine Umfangserweiterung in den Bereichen Haustechnik, Facility Management (FM), Kosten- und Bauablaufplanung fand über das Release 2.0 aus dem Jahr 1999 statt. Da die Qualitätssicherung der IFC 2.0 Spezifikationen aufgrund von Zeitzwängen bei der Entwicklung nicht ausreichend sicher gestellt werden konnte, wurde bereits im Jahr 2000 die Entwicklung der Nachfolgeversion IFC 2x beschlossen. Diese wurde im Oktober desselben Jahres veröffentlicht. Der Umfang des IFC-Modells wurde mit der Einführung neuer Anwendungsbereiche und Konzepte stark ausgeweitet. Diese Erweiterungen führten zu einer Veränderung des IFC Core Layers, bei dem die Kompatibilität des Release 2x zu dem vorangegangenen Release 1.5.1 nicht mehr sichergestellt werden konnte.

Die *IFC 2x Plattform* wurde im November 2002 als ISO 16739 registriert und über die IFC Release 2x Addendum 1 (Oktober 2001), die IFC Release 2x2 (Mai 2003) und die aktuelle IFC Release 2x3 (IFC Release 2x Addendum 2) vom Februar 2006 weiter verbessert. Die Schnittstellen von Produkten führender Softwarehäuser wie beispielsweise NEMETSCHEK, GRAPHISOFT, TEKLA, BENTLEY, SOLIBRI und AUTODESK basieren auf den Spezifikationen der IFC 2x Plattform [93]. Für detaillierte Informationen zu den Releases bis Version 2x2 wird auf [67, 95, 161] und zur aktuellen Version 2x3 auf [93, 97] verwiesen.

2.4 Das Autodesk AEC Produktmodell

Es besteht ein grundsätzlicher Konflikt zwischen der IAI und den Entwicklern der CAD-Systeme. Während sich die IAI bemüht, eine Standardisierung der IFC für einen konfliktfreien Austausch zwischen allen auf dem Markt befindlichen Applikationen zu gewährleisten, geben Systemhersteller ihren Produkten und integrierten Datenmodellen besondere Merkmale, um sich von der Konkurrenz abzuheben und einen Marktvorteil zu erlangen. Dieser Ansatz resultiert in der Entwicklung individueller Insellösungen.

Gerade das Softwarehaus AUTODESK ist sich aktuell mit den Entwicklern der IAI in vielen Punkten uneinig und baut aufgrund seiner Marktführungsposition einen „Standard“ über das haus eigene AEC Produktmodell auf. Dieses steht in direkter Konkurrenz zu den IFC. Das AUTODESK Produktmodell findet bis zum Jahr 2007 Anwendung in den Produkten *Architectural Desktop* [16, 24] sowie in dem auf einer parametrischen Gebäudemodellierung [23] basierenden *Revit Building*¹³ [22]. Entgegen der Definition „AEC“ (Architecture, Engineering and Construction) sind nur AUTODESK-Produkte, welche ein Produktmodell unterstützen, dem AEC-Bereich zugeordnet. Die bekannte Software *AutoCAD* [18] fällt somit nicht unter diesen Bereich¹⁴.

AUTODESK ist eines der Gründungsmitglieder der IAI und ebenfalls in fast allen IAI-Chaptern vertreten. Daher verwundert es nicht, dass das im Architectural Desktop (ADT) integrierte Produktmodell Ähnlichkeiten zu dem IFC Produktmodell aufweist. Bis zur Version 3.3 enthielt das ADT eine Schnittstelle für den Import und Export von IFC Produktmodellen, die auf den Standards IFC 1.5.1 und 2x basieren. In den nachfolgenden Versionen wurde diese Funktion entfernt, obwohl ein Datenaustausch bezogen auf die technischen Aspekte weiterhin möglich ist. Dieser Mangel der aktuellen ADT-Versionen wird über das Plug-In [80] der Firma INOPSO zur Erweiterung der Applikationen um eine IFC-Schnittstelle beseitigt.

Das ADT stellt über sein Produktmodell vielfältige Funktionen zur Erstellung und Bearbeitung von Bauteilen sowie zur Bauteilassemblierung zur Verfügung. Die Entwicklung spezifischer Stile von Bauteilarten und das Anbinden zusätzlicher Objekteigenschaften über statische und dynamischen Listen (Eigenschaftssätze oder *Property Sets* bzw. *Automatic Property Definitions* [17]) ist möglich. Über das eingesetzte INOPSO Plug-In ist eine konsistente Überführung dieser Eigenschaftserweiterungen in ein IFC Modell gewährleistet. Die Abbildung 2.8 stellt Eigenschaftssätze von Bauteilen eines AEC Produktmodells dar.

¹³ REVIT TECHNOLOGY wurde Ende 1997 gegründet. AUTODESK selber beschreibt das Produkt Revit als Planungs- und Dokumentationssystem. Im Gegensatz zu den AUTODESK-eigenen CAD-Anwendungen sowie auch CAD-Applikationen anderer großer Softwarehäuser, deren Basisquellcodes recht alt sind und das Produktmodell somit über die Zeit um das geometrisch orientierte CAD-Grundgerüst herum implementiert wurde, ist Revit eine vollständige Neuentwicklung. Der Fokus bei der Entwicklung lag von Beginn an auf der Idee des *Building Information Modeling* (BIM).

¹⁴ Im Jahr 2007 erfolgte die Umbenennung einiger AUTODESK Produkte. Die aktuelle Version 2008 des *Architectural Desktops* wurde in die *AutoCAD* Produktfamilie aufgenommen und wird nun unter dem Namen *AutoCAD Architecture 2008* kommuniziert. Da das Produkt *AutoCAD 2008* weiterhin kein Produktmodell unterstützt, behält die Produktzuordnung in den AEC-Bereich ihre Gültigkeit.

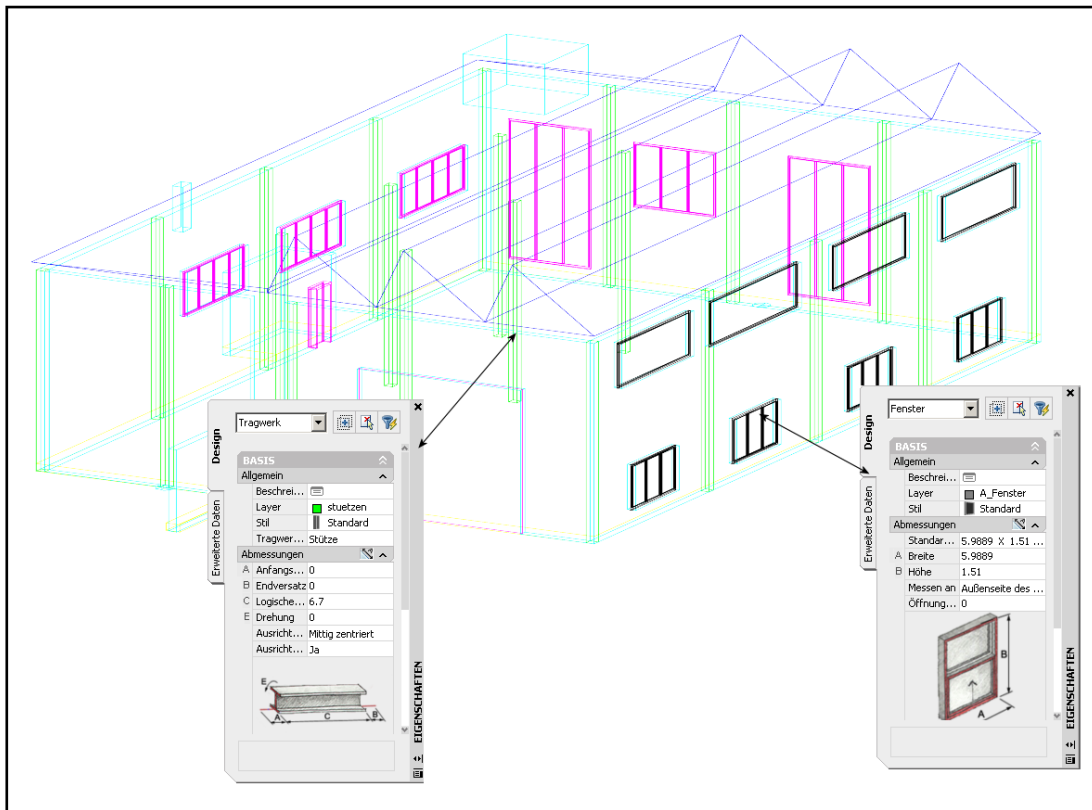


Abbildung 2.8: AEC Produktmodell im ADT mit bauteilspezifischen Metadaten (Property Sets)

Die Zeichnungsebenen des ADT bleiben bei einer Konvertierung in ein IFC Produktmodell erhalten. IFC-Objekte, deren Interpretation bei einem Importvorgang über das ADT nicht möglich ist, werden als Platzhalterobjekte (*Proxy Elements*) abgebildet. Das ADT bietet die Möglichkeit der Modellierung von IFC Objektklassen, welche als nicht standardisiertes Bauteil vom Produktmodell des ADT unterstützt werden. Hierbei können beliebige ADT Massenelemente mit einem Typisierungsattribut (beispielsweise IFCBuilding, IFCSpace) versehen und als IFCBuildingElementProxy exportiert werden. Für weiterführende Informationen zu dem Produktmodell des ADT wird auf das Kapitel 3 und entsprechende Abschnitte von Kapitel 6 verwiesen.

2.5 Schlussfolgerung und Ausblick

In der Praxis stößt das wenig erprobte IFC Produktmodell aufgrund geringer Erfahrungswerte auf Ablehnung, da Nutzen und Vorteile der Interoperabilität nicht im Verhältnis zu Risiko und Mehraufwand stehen. Die Mehrkosten für Unternehmen ergeben sich aus den höheren Wissensanforderungen an den Nutzer und dem höheren Zeitaufwand für die Modellerstellung. Verstärkt wird dieser Effekt durch die nicht ausreichend geregelten Bestimmungen der *Honorarverordnung für Architekten und Ingenieure HOAI* [188], bei der die Erstattung von Unkosten für eine Produktmodellerstellung im Allgemeinen nicht berücksichtigt wird. Im Bereich der Forschung und Wissenschaft beschreiben Experten die IFC als Produktmodell in einem „hauptsächlich prototypischen Stadium“ [67].

Entgegen dieser nüchternen Sichtweise hinsichtlich der IFC sind einige Pioniere durchaus bereit, Risiken einzugehen und Großprojekte unter Verwendung des IFC Modells als Planungsgrundlage durchzuführen, wie beispielsweise das Projekt *Umbau des Hauptbahnhofs München*¹⁵ [163] oder die *3D-Fabrikplanung bei der AUDI AG Ingolstadt* [30]. Der Erfahrungsbericht bei dem Projekt AUDI zeigt, dass ein Produktmodell im jetzigen Entwicklungsstand durchaus das Potential besitzt, als Planungsgrundlage zu dienen und die Koordination sowie den Datenaustausch zwischen den Fachplanern merklich zu verbessern. Es wurden aber auch Schwächen der IFC hinsichtlich mangelnder Geometriehandhabung sowie der nicht hinreichenden Schnittstellenimplementierungen verschiedener CAD Systeme aufgezeigt [31].

Die Bauindustrie bewegt sich nicht als Ganzes. Eine Umstellung ist ein großer konzeptioneller Sprung. Hierzu fehlen allgemeingültige Standards, wie sie in der Automobilindustrie über STEP existieren. Diese sind im Bauwesen nur länderspezifisch festgelegt. Schon die Umstellung von einer zweidimensionalen Konstruktionszeichnung zu einem dreidimensionalen Geometriemodell ohne die Berücksichtigung von Metainformationen hat global noch nicht stattgefunden. Dass die Softwareindustrie bereits angemessene Funktionalitäten über ihre Applikationen bereitstellt, bleibt von den Anwendern weitgehend unbeachtet. Insgesamt ist daher eine kurzfristige Etablierung des IFC-Standards nicht zu erwarten. Nach einer Prognose von PHILLIP BERNSTEIN, dem Vice President Building Solution Division der Firma AUTODESK, beschleunigt sich aber die Bewegung in Richtung der Anwendung von IFC und dreidimensionaler Modelle [28].

Neben den Bemühungen zur Etablierung eines standardisierten Produktmodells, existiert im Bereich des Bauwesens der Bedarf für ein allgemeingültiges Format zum Austausch von Gebäudedaten. Eine Lösungsmöglichkeit stellen die Ansätze der Unternehmung ADOBE dar. Hierzu wurde die Softwarefamilie *Acrobat* [3] erweitert. Diese gilt als weltweiter Marktführer für den Austausch von Dokumenten über das *PDF-Format*. Durch die Einführung eines AEC Bereiches [1, 2] sowie über das Produkt *Acrobat 3D* [4] soll ein kooperativer Datenaustausch im Bereich des Bauwesens ermöglicht werden. Die neuen Produkte erlauben die Einbindung dreidimensionaler Modelle mit Zeichnungsebenen,

¹⁵ Ausschreibungsgrundlage war ein Modell im IFC2x2 Format

Maßstabsangaben und objektbezogenen Textinformationen über das PDF-Format. Der Benutzer erhält die Möglichkeit, Modelle von Bauwerken und Bauwerkselementen in einer dreidimensionalen Ansicht zu betrachten und diese über eine Navigation zu drehen und zu vergrößern. Darüberhinaus kann er Animationen, wie beispielsweise Raumluftrömungen, abspielen. Die Firma NEMETSCHEK hat als erstes Softwarehaus für CAD Anwendungen reagiert und ist eine Partnerschaft mit ADOBE zur Implementierung von Schnittstellen für den Im- und Export von PDF-Dateien für die Produkte *Allplan*, *Cinema4D* und *VectorWorks* eingegangen [158].

Dieser Weg kann als wichtiger Zwischenschritt gewertet werden, da das *Acrobat* Format als Standard für ein Dokumentenaustauschformat gilt, mit dem zugrunde liegenden Datenmodell allerdings nicht die notwendige Komplexität eines Produktmodells erfüllt.

Kapitel 3

Der virtuelle Entwurfsraum

3.1 Der ADT als Modellierer

Der Architectural Desktop 2005 (ADT) der Unternehmung AUTODESK dient als Basis für den *virtuellen Konstruktionsraum* des Prototyps. Die Mehrheit der projektbezogenen Funktionalitäten wurde hierüber realisiert. Diese beinhalten Kommunikationsschnittstellen, Überwachungswerkzeuge für den Konstruktionsprozess, Teile des *Pre-Processing* sowie die konsistente Integration des Produktmodells.

Der ADT, eine Erweiterung der Software AutoCAD [18], ist ein *De-facto-Standard* im Bauingenieurwesen und bietet neben leistungsfähigen Modellierungsfunktionalitäten auch umfassende, objekt-orientierte Programmierschnittstellen. Hierzu zählt das *Object Modeling Framework* (OMF) [15]. Diese Schnittstelle, eine Erweiterung der Schnittstelle *Autodesk ObjectARX* [19], bietet einen effizienten Datenaustausch mit den Modulen des Konstruktionsprozesses, einen Zugriff auf Geometrie und Metainformationen der Bauteilelemente, sowie die Kopplung an das AUTODESK-eigene AEC Produktmodell [15, 17].

3.2 Schnittstellen des ADT

Der AUTODESK Architectural Desktop (ADT) bietet eine Anzahl leistungsfähiger Programmierschnittstellen. Als Beispiele seien *AutoLISP* und *VisualLISP*, *Visual Basic*, *COM* und *ActiveX* genannt, sowie die zur Entwicklung integrierter Zusatzfunktionen des Prototyps genutzten Schnittstellen *ObjectARX*, *ObjectDBX* und das *Object Modeling Framework* (OMF).

Die Makrosprache **AutoLISP** und deren Erweiterung **VisualLISP** dienen der Automatisierung von Abläufen innerhalb des ADT. Ihr Funktionsumfang beschränkt sich auf eine sequenzielle Abfolge von Aufrufen bestehender ADT-Methoden. AutoLISP ist eine Implementierung von *Common Lisp* [204] und benötigt keinen Compiler. VisualLISP, welches seit 2000 in AutoCAD und den ADT integriert ist, erweitert AutoLISP um eine Entwicklungsumgebung mit einer automatisierten Syntax-Kontrolle und einem Datei-Compiler [191].

Die von MICROSOFT entwickelte **COM+/ActiveX**¹⁶ [146, 150] Schnittstelle erlaubt die Steuerung des ADTs innerhalb der MICROSOFT *Windows* Plattform [149] über beliebige Anwendungen, welche ihrerseits eine COM-Schnittstelle besitzen. Ein Client-Server-Prinzip ermöglicht den Aufruf von Methoden und die Übermittlung von Daten über die ActiveX-Schnittstelle des ADT (*Object Model Interface, OMI*). Durch die bidirektionale Funktionsweise der Schnittstelle können COM/ActiveX unterstützende Windows-Applikationen auf einfache Weise verknüpft werden. Die Kopplung des *HVAC-Agenten Frameworks* mit dem ADT wurde über diese Schnittstellentechnik realisiert.

3.2.1 ObjectARX

Die objektorientierte Schnittstelle ObjectARX [19] basiert auf der Sprache C++ [108] und wird neben dem ADT auch in AutoCAD eingesetzt. Ihre Einführung erfolgte in der Version 13 und löste die auf der Sprache C [103] basierende Schnittstelle *ADS* ab. Die Programmierung der AutoCAD Applikation wurde von AUTODESK mit ObjectARX realisiert.

Über ObjectARX erstellte Applikationen (ARX-Module) haben Entwickler die Möglichkeit, auf Grundfunktionen von AutoCAD (Zeichenebene, Zeichendateien, grafische Programmoberfläche, Datenbanken) zuzugreifen und Modifikationen an geometrischen Objekten durchzuführen. Die Funktionen können über eine von AUTODESK zur Verfügung gestellte C++ Klassenbibliothek durch den Ansatz der objektorientierten Programmierung erweitert werden [126, 141]. Die Tabelle 3.1 führt eine Auswahl wichtiger ObjectARX-Klassen auf.

AcRx	Klassen zum Binden der Applikation und zur Registrierung und Identifikation in der Laufzeitumgebung
AcEd	Klassen zur Registrierung einfacher AutoCAD Befehle
AcDb	AutoCAD Datenbank Klassen
AcGi	Klassen zum Rendern grafischer AutoCAD Objekte
AcGe	Allgemeine Klassen für geometrische Objekte

Tabelle 3.1: Die ObjectARX Klassenbibliothek (Auszug)

Beispiel für eine Klassendefinition ist die Klasse `AcGePoint3d` zur Darstellung eines Punktes als geometrisches Objekt mit 3 Raumkoordinaten. `AcDbObjectId` definiert eine Klasse zur Zuweisung einer eindeutigen Objekt-Identifikationsnummer innerhalb der AutoCAD Datenbank.

Als Entwicklungsumgebung von ObjectARX wird das über ein AUTODESK Add-In [20] erweiterte *Visual Studio .NET* von MICROSOFT eingesetzt [145]. ARX-Module können während der Laufzeit der CAD-Applikation (ADT und AutoCAD) dynamisch geladen und entladen werden.

¹⁶ ActiveX bezeichnet ein Softwarekomponenten-Modell von MICROSOFT für aktive Inhalte. ActiveX-Komponenten erweitern die Component Object Model-Standards (COM) von MICROSOFT.

3.2.2 Das Object Modeling Framework

Das AUTODESK *AEC Object Modeling Framework (OMF)* [15] stellt eine Erweiterung der ObjectARX Schnittstelle dar und dient der Entwicklung bauteilorientierter AEC-Objekte für den AUTODESK Architectural Desktop. Das OMF unterstützt als einzige von AUTODESK implementierte Schnittstelle ein Produktmodell. Die im Rahmen dieser Arbeit entwickelten Funktionen des virtuellen Entwurfsraums wurden hinsichtlich der Konsistenz zu dem IFC bzw. AEC Produktmodell über das OMF realisiert.

Analog der ObjectARX Schnittstelle, mit welcher die Applikation AutoCAD entwickelt wurde, diente das Object Modeling Framework der Entwicklung des AUTODESK Architectural Desktops. Das ADT ist somit eine OMF Erweiterung der auf ObjektARX basierenden AutoCAD Applikation. Die Zusammenhänge der beiden Applikationen und Schnittstellen werden in Abbildung 3.1 wiedergegeben.

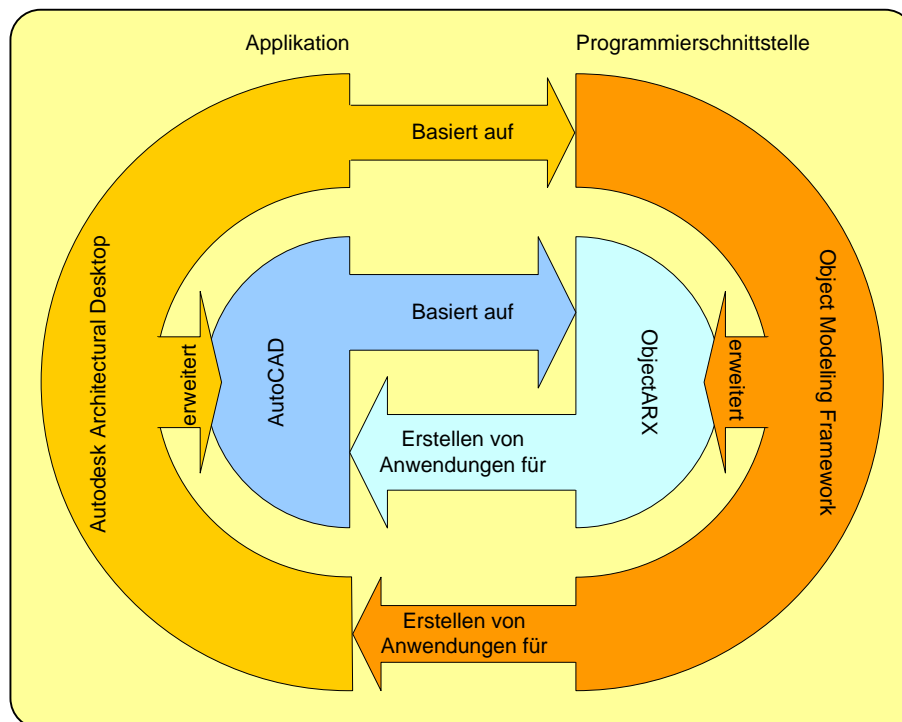


Abbildung 3.1: Darstellung der Beziehung zwischen den Applikationen AutoCAD und Architectural Desktop und den Programmierschnittstellen ObjectARX und Object Modeling Framework (OMF).

Entwickler erhalten bei der Programmierung von OMF-Anwendungen über eine C++ Klassenbibliothek den Zugang zur AUTODESK-eigenen AEC Kerntechnologie. Über diese erfolgt ein Zugriff auf Bauteile, Bauteilgruppen, Baustile, Objektbeziehungen und spezifische Metainformationen von Objekten des Produktmodells. Die Klassen der ObjectARX Bibliothek werden für den Zugriff auf nicht-produktmodellbezogene geometrische Funktionen benötigt (siehe Abbildung 3.2).

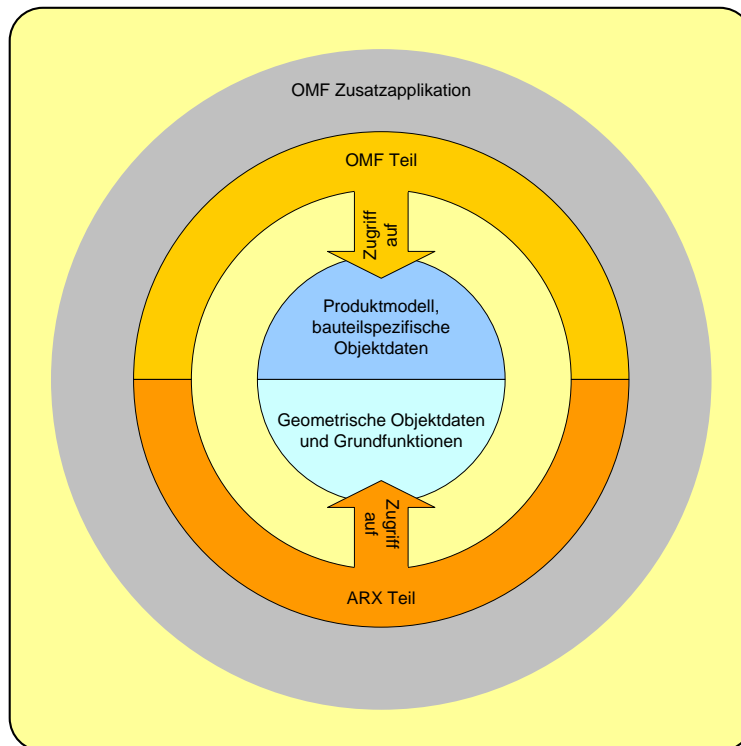


Abbildung 3.2: Bei der Entwicklung von OMF Zusatzapplikationen werden OMF-Klassen für den Zugriff auf bauteilspezifische Objekte des Produktmodells genutzt. Zusätzlich werden die der Struktur des OMF zugehörigen ObjectARX-Klassen für die Kontrolle von grundlegenden geometrischen Objekten und Zeichenfunktionen benötigt.

Die ObjectARX-Klassen bilden ebenfalls die Schnittstelle zur Laufzeitumgebung der Anwendung und zu Datenbanken. OMF-Zusatzapplikationen können analog zu den ARX-Modulen zur Laufzeit des ADT dynamisch geladen und entladen werden.

3.2.2.1 Interoperabilität des Produktmodells

Bei der Erweiterung eines Produktmodells über das Object Modeling Framework um benutzerspezifische Metainformationen sind konzeptionelle Ansätze zur Gewährleistung der Interoperabilität erforderlich. Diese gehen mit dem Prinzip der objektorientierten Programmierung nicht konform.

Daten werden über einen objektorientierten Ansatz durch das Ableiten von Klassen und die Deklaration zusätzlicher Attribute und Methoden erweitert. Die Ableitung eines Bauteiles, beispielsweise einer Tür, bewirkt die Generierung eines neuen Bauteiltyps mit zusätzlichen Attributen. Dieses Objekt kann bei einer Konvertierung und Weiterverarbeitung über das ADT nicht mehr interpretiert werden und wird nicht als Bauteil des Produktmodells verarbeitet. Das Ableiten von Klassen ist im OMF möglich, führt aber bei produktmodellspezifischen Anwendungen zu Inkompatibilitäten.

Das OMF sieht hierbei die Erweiterung gegebener Bauteile und Stile über spezifische Bauteileigenschaften (*Properties*) vor, welche in Form von Eigenschaftssätzen (*Property Sets*) mit dem Objekt verknüpft werden. Das ADT bietet die Möglichkeit der Darstellung und

Modifikation dieser Eigenschaftssätze über eine grafische Oberfläche¹⁷ (GUI) [147], welche eine oder mehrere Eigenschaftslisten¹⁸ eines Objektes in tabellarischer Form darstellt (siehe hierzu auch Abbildung 2.8).

3.2.2.2 OMF Klassenhierarchie eines AEC Bauteils

Eine besondere Eigenschaft des OMF besteht in der Teilung der Klassenhierarchie der AEC Objekte. Hierbei wird ein Bauteil des AEC Produktmodells in die beiden Basisklassen `AecDbGeo` und `AecDbDictRecord` aufgeteilt. `AecDbGeo` dient dem vorhalten physikalischer Bauteilinformationen wie Breite, Höhe oder Position. `AecDbDictRecord` beschreibt Informationen zum Stil des Bauteils wie Form, Typ oder Materialeigenschaften. Als Beispiel sei das Bauteil „Wand“ genannt: Die Klasse `AecDbWall`, abgeleitet von `AecDbGeo`, besitzt spezifische Bauteilinformationen. Übergeordnete Metainformationen, welche mehreren Wandobjekten zugeordnet werden können, sind in der Klasse `AecDbWallStyle` (Ableitung der Klasse `AecDbDictRecord`) gekapselt (siehe Abbildung 3.3).

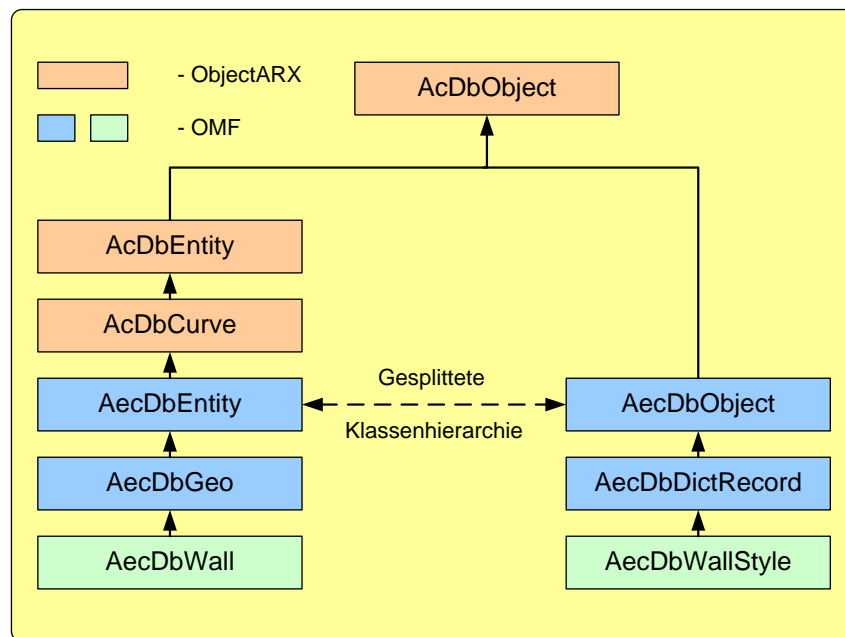


Abbildung 3.3: Teilung der physikalischen Informationen bei der OMF Klassenhierarchie in AEC Bauteil- und Stilobjekte. Die übergeordneten, dem ObjectARX zugehörigen Basisklassen dienen der Zuordnung der Bauteilelemente zum AutoCAD-Grundsystem.

¹⁷ Das OMF unterstützt die Erstellung eigener grafischer Oberflächen über die *Microsoft Foundation Classes (MFC)*, eine Sammlung objektorientierter Klassenbibliotheken, die von MICROSOFT speziell für die Programmierung von Anwendungen mit grafischen Benutzeroberflächen unter MICROSOFT Windows auf Basis der Sprache C++ entwickelt wurde. Alle grafischen Oberflächen des AUTODESK Architectural Desktops sowie von AutoCAD wurden ebenfalls mit den MFC realisiert.

¹⁸ Es ist generell möglich, innerhalb des ADT's ein AEC Objekt mit beliebig vielen Eigenschaftslisten (Property Sets) zu verknüpfen, welche wiederum eine beliebig große Menge von Attributen (Properties) besitzen.

Dieses Konzept bietet für die Erweiterung des Bauwerkmodells gute Möglichkeiten, komplexe Änderungen von Stilen auf einfache Weise durchzuführen oder neue Bauteile einzuführen und mit vorhandenen Stileigenschaften zu verknüpfen. Eine gleichzeitige Interoperabilität zum Produktmodell ist dabei gewährleistet. Die Wandinstallation eines Klimaanlageeinlasses hat beispielsweise ähnliche Verhaltenscharakteristiken wie ein Fenster. Die Positionierung einer solchen Installation erfolgt immer an dem Objekt Wand bei identischer Ausprägung von Eigenschaften wie Höhe und Breite. Bei der Übernahme der Stileigenschaften für das Objekt „Fenster“ auf das neue Objekt „Klimaanlageeinlass“ können somit komplexe Attribute wie Anker Elemente und grafische Darstellungsformen direkt auf das Klimaanlageelement übertragen werden. Eine Neuerstellung ist hierbei nicht notwendig.

Eine weitere Trennung über das OMF erfolgt bei der Datenhaltung physikalischer und grafischer Eigenschaften eines Bauteilobjektes. Für die grafische Ausgabe definiert das OMF über die *Display Representations* eigene Objekte, welche die Visualisierung des Bauteils in unterschiedlichen Ansichten ermöglichen. Das OMF unterstützt die Kopplung beliebiger Display Representations (*AecDbDispRep*) an ein Bauteilelement (*ADT Entity/AecDbGeo*). Display Representations werden als abstrakte Klasse ohne grafische Informationen vorgehalten. Die grafischen Daten sind über *Display Properties* (*AecDbDispProp*) an jeweils eine Display Representation gebunden (siehe Abbildung 3.4).

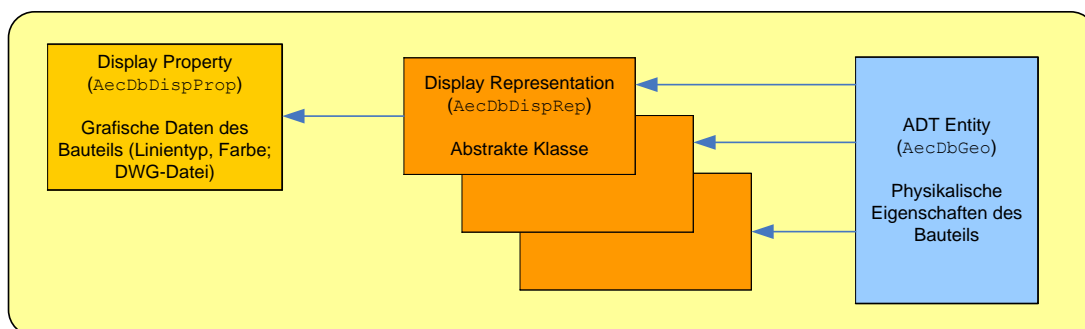


Abbildung 3.4: Kopplung verschiedener grafischer Darstellungen über Display Representations und Display Properties an ein AEC Bauteilelement.

Die Display Properties enthalten die Daten des Bauteilelementes wie Farbe, Linientyp, und Füllung und werden über eine *DWG-Datei* [16, 17] vorgehalten. Das ADT benutzt eine Standarddarstellung des Bauteilelementes, falls keine eigene Display Property definiert ist. Auch bei Stilobjekten hat diese Darstellungsdefinition Gültigkeit.

Der Ansatz ermöglicht die fallspezifische Zuordnung einer beliebigen Anzahl von grafischen Darstellungsarten für ein Bauteilelement. Dies ermöglicht die Selektion verschiedener Planungsansichten, wie beispielsweise der Grundrissdarstellung, der Deckenplandarstellung oder der in Abbildung 3.5 dargestellten Isometrie. Die Realisierung länderspezifischer oder fachplanerspezifischer Darstellungsformen ist hierüber ebenfalls möglich.

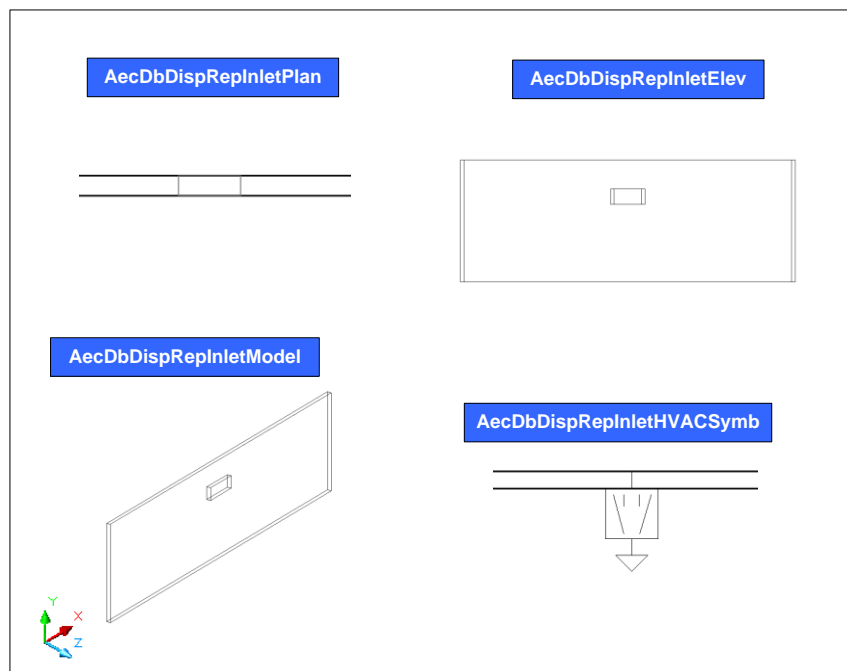


Abbildung 3.5: Die grafische Darstellung eines Bauteilelementes kann über eine beliebige Anzahl geometrischer Attribute an ein AEC Objekt angehängt werden und wird abhängig vom spezifischen Anwendungsfall über unterschiedliche „Display Representations“ dargestellt.

Die zur Optimierung des Raumklimas im Rahmen des Simulationsbeispiels beteiligten Fachplaner Architekt, Innenarchitekt und Hausleittechniker benötigen entsprechend ihrer spezifischen Aufgabenstellung Baupläne desselben Planungsvorhabens in unterschiedlichen Ausführungsformen. Der Architekt ist bei einem Klimaanlageneinlass nur an den statischen Eigenschaften interessiert und benötigt eine Darstellung des Wanddurchbruchs für die Klimaanlageninstallation. Der Hausleittechniker benötigt hingegen einen Bauplan des Klimaanlageneinlasses als symbolische Darstellung einer regeltechnischen Anlage (wie in der Abbildung 3.5 als `AecDbDispRepInletHVACSymb` dargestellt). Die geometrischen Eigenschaften des Bauwerks haben für den Hausleittechniker keine Relevanz. Durch die Zuordnung unterschiedlicher Display Representations für das Objekt „Klimaanlageneinlass“ und einer automatisierten Filterung zur Erstellung unterschiedlicher Ansichten ist eine einfache Generierung von fachspezifischen Bauplänen auf der Basis eines Datenmodells möglich. Funktionen wie der in Abschnitt 3.3.4 beschriebene *LayoutManager* wurden durch Nutzung dieser Techniken realisiert.

3.3 Funktionen des virtuellen Entwurfsraums

Die Abbildung 3.6 beschreibt die Funktionen des virtuellen Entwurfsraums, welche im Rahmen dieser Arbeit in den ADT implementiert wurden.

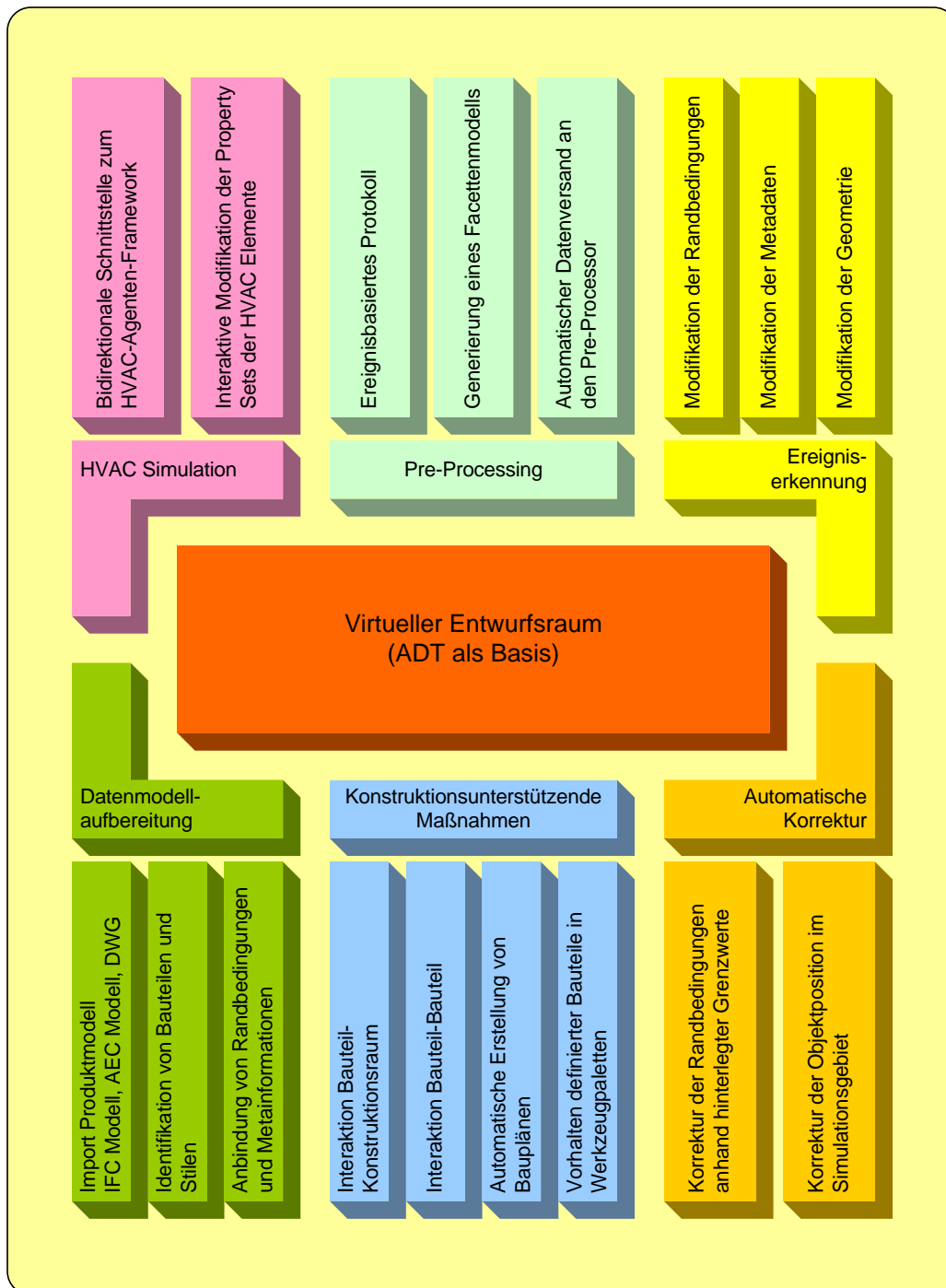


Abbildung 3.6: Funktionen des virtuellen Entwurfsraums

Die Erweiterung des ADT erfolgt über das Modul `DesignTriActor`, welches als OMF-Applikation in die Laufzeitumgebung des ADT eingebunden wird. Die Funktionsweisen seien an dieser Stelle zusammenfassend erläutert und werden in den nachfolgenden Abschnitten dieses Kapitels genauer dargestellt.

Ereigniserkennung und automatische Korrektur. Mehrere Überwachungsroutrinen (*Reactor* [19, 141]) sind während des Konstruktionsprozesses aktiv und kontrollieren kontinuierlich den virtuellen Entwurfsraum auf Inkonsistenzen. Hierbei werden geometrische Modifikationen von Objekten und Eingaben von Randbedingungen für die HVAC und CFD- Simulation berücksichtigt. Befindet sich ein Objekt beispielsweise außerhalb des Simulationsgebietes, erfolgt eine Warnung an den Benutzer mit einer anschließenden automatischen Verschiebung des Objektes auf eine korrekte Position. Werte für objektbezogene und globale Randbedingungen der Simulation werden bei ihrer Eingabe mit Grenzwerten validiert. Die spezifischen Grenzwerte sind in einer *XML-Datenbank* (siehe [27, 218] und Abschnitt 6.4.2) hinterlegt. Bei Überschreitung der Minimal- oder Maximalwerte erfolgt eine automatische Korrektur auf den definierten Grenzbereich.

Konstruktionsunterstützende Maßnahmen. Das geometrische Verhalten zwischen zwei Objekten wird über eine Kopplung an *Ankerelemente* [15, 17] realisiert. Hierüber können beispielsweise Abhängigkeiten zwischen einem Klimateinlass und einer Wand oder einer Bodenplatte, einer Tür und einer Wand oder einer Tür und einer Tür abgebildet werden. Neben einer Bauteil-Bauteil-Verknüpfung ist zusätzlich die Definition von Verhaltensweisen eines Bauteils in Bezug auf den globalen Konstruktionsraum möglich: Eine Steckdose soll nur in einer Höhe von 30 cm über dem Boden und nur an dem Objekt „Wand“ montiert werden. Der Mindestabstand zwischen zwei Wänden soll zwei Meter nicht unterschreiten.

Ein Ankerelement ist als Punkt, Linie, Fläche oder Volumen definiert und stellt über seine geometrischen Dimensionen die Bewegungsgrenzen für das verknüpfte Objekt dar. Bezogen auf das erste Beispiel ist das Ankerelement als horizontale Linie definiert, welche mit dem Objekt „Wand“ verbunden ist und die Bewegungsfreiheit für das verknüpfte Objekt „Steckdose“ bestimmt.

Das ADT hält definierte Bauteile eines Produktmodells wie beispielsweise Wände, Türen und Fenster als AEC-Objekte in Objektbibliotheken (Werkzeugpaletten [17]) vor. Mit Hilfe dieser Paletten können Objekte intuitiv vom Benutzer dem Konstruktionsraum hinzugefügt werden. Die Steuerung erfolgt hierbei über eine einfache Verschiebung des Objektes aus der Palette an die gewünschte Position. Zur Durchführung einer HVAC- und CFD-Simulation werden zusätzliche Konstruktionsobjekte wie Ein- und Auslässe einer Klimaanlage, Messsensoren, simulationsspezifische Bauteile sowie Funktionen zur Steuerung des Simulationsprozesses benötigt. Alle Erweiterungen wurden über das OMF als AEC-Objekte implementiert und sind somit in Werkzeugpaletten integrierbar (siehe Abbildung 3.7).

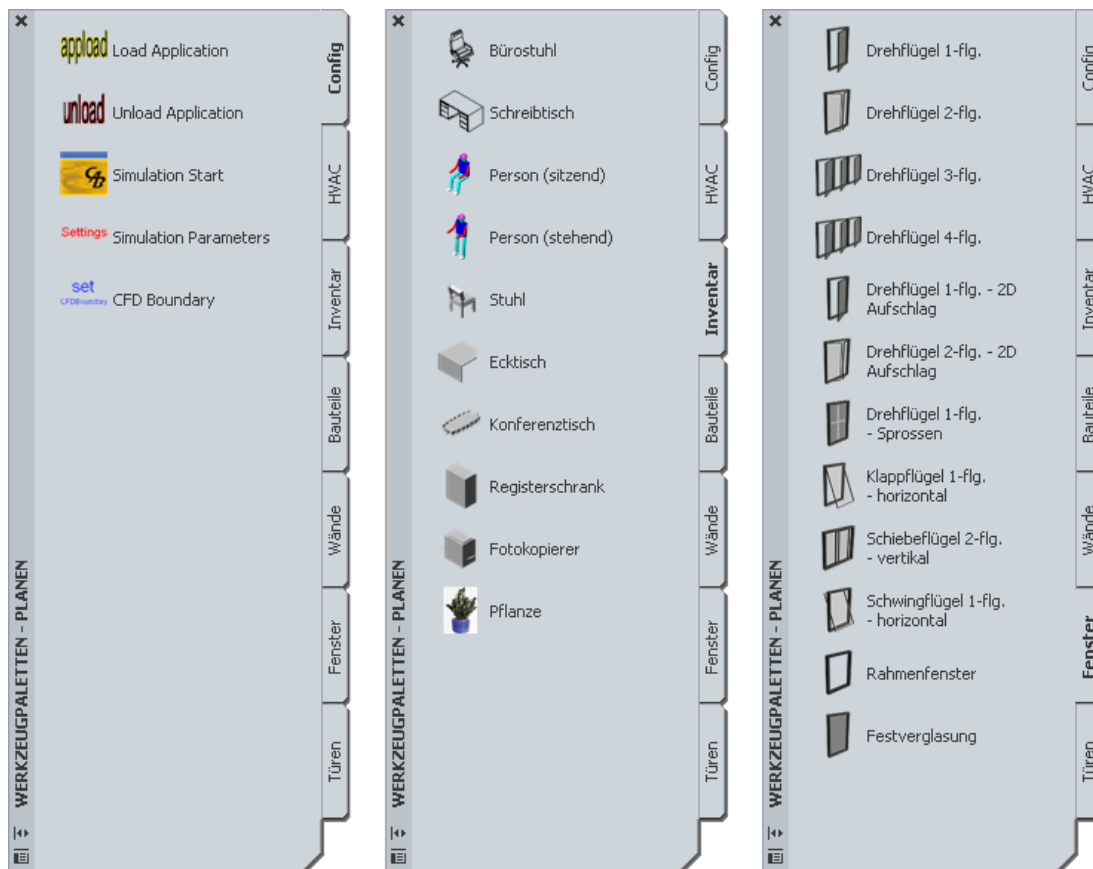


Abbildung 3.7: Erweiterung der ADT-Werkzeugpaletten um eigene Funktionen und Elemente. Durch die Erstellung von AEC-Objekten mit Hilfe des OMF ergibt sich die Möglichkeit der Erweiterung vorhandener Werkzeugpaletten des ADT um Funktionen zur globalen Steuerung des Prototyps (links) sowie eigener Bauteilelemente mit spezifischen Randbedingungen (Mitte). Diese werden zusammen mit ADT-eigenen Bauteilen (rechts) in einer Werkzeugpalette vorgehalten.

Durch diesen Ansatz konnte auf eine Implementierung zusätzlicher grafischer Oberflächen in das CAD-Programm verzichtet werden. Die gewohnte Benutzerführung des Programms bleibt hierdurch für den Anwender erhalten.

Eine konstruktionsunterstützende Maßnahme bietet die Funktionserweiterung zur automatischen Erstellung von Bauplänen für spezifische Fachplaner. Über den objektorientierten Ansatz des Produktmodells werden Bauteilen fachplanerspezifische Filterattribute zugewiesen. Bei einer automatisierten Plangenerierung werden die über Attribute selektierten Bauteile an den jeweiligen spezifischen Bauplan übertragen. Die Weiterverarbeitung der Pläne kann neben der Ausgabe auf einen Plotter auch über digitale Formate (wie beispielsweise ADOBE Acrobat PDF [3]) erfolgen. Ein Onlinezugriff auf die Plandaten ist über einen angebundenen Webserver möglich. Die genaue Beschreibung der automatischen Bauplanerstellung wird in Abschnitt 3.3.4 weitergeführt.

Datenmodellaufbereitung. Der Modellierer erlaubt den initialen Datenimport der Konstruktionszeichnung über ein IFC oder AEC Produktmodell. Hierbei erfolgt eine automatische Aufbereitung der Daten für den Simulationsvorgang. Beide Produktmodelle unterstützen keine Implementierung von Randbedingungen für CFD-Simulationen und notwendiger Parameter und Objektattribute für den interaktiven Simulator. Erforderlich sind beispielsweise Randbedingungen von Temperatur und Geschwindigkeit der eingeblasenen Luft sowie Druck, Strahlung und Materialeigenschaften. Weiterhin sind Parameter für Regelglieder, Definitionen zu Grenzwerten und Attribute zur fachplanerspezifischen Identifikation von Bauteilen erforderlich. Diese Zusatzinformationen werden über Eigenschaftssätze (Property Sets¹⁹) mit den Objekten verknüpft. Die Datenaufbereitung des Produktmodells erfolgt automatisch mit dem Importvorgang über den Modellierer. Hierbei werden Objektattribute auf ihre Vollständigkeit überprüft und hinterlegte Werte mit definierten Grenzwerten verglichen. Treten Abweichungen auf, werden automatisch neue Werte gesetzt und fehlende Attribute oder fehlende Eigenschaftssätze hinzugefügt. Dabei greifen die Kontrollroutinen auf Standardvorgabewerte zurück, welche in einer *XML-Datenbank* [27, 218] hinterlegt sind. Die Identifizierung des importierten Objektes erfolgt über die produktmodellspezifische Typ- und Stilzuordnung.

HVAC Simulation. Für die interaktive Simulation unter Berücksichtigung von Mess-, Steuer- und Regeltechniken werden Software-Agenten des *HVAC-Agenten Frameworks* über eine *bidirektionale COM-Schnittstelle* [146] an den geometrischen Modellierer angebunden. Der Modellierer überwacht geometrische Modifikationen innerhalb des Entwurfsraums und übermittelt relevante Informationen an das Agenten Framework. Entsprechende Änderungen des zur Laufzeit über die Agenten berechneten Luftstromes der Klimaanlageeinlässe sendet das HVAC-Agenten Framework über die COM-Schnittstelle an den Modellierer. Die übermittelten Informationen werden als neue Randbedingungen für den CFD-Kern im Eigenschaftssatz des jeweiligen HVAC Elementes gesetzt und periodisch aktualisiert. Für eine detaillierte Beschreibung des HVAC-Agenten Frameworks wird auf das Kapitel 4 verwiesen.

Pre-Processing. Teilprozesse zur Aufbereitung des Produktmodells für die numerische Simulation werden innerhalb des Modellierers durchgeführt. Ereignisgesteuerte Protokolle initiieren die automatische Aufbereitung und Weiterverarbeitung der Simulationsdaten. Erfolgen simulationsrelevante geometrische Modifikationen innerhalb des Konstruktionsraumes (Verschieben, Hinzufügen oder Löschen von Objekten), wird die Geometrie über einen Hintergrundprozess des Modellierers in ein *volumenorientiertes Facettenmodell* (*Brep-Modell* [16, 161]) konvertiert. Die Übertragung an die zur Weiterverarbeitung zuständigen Module auf dem PC-Cluster erfolgt über eine *Socket-Schnittstelle* [208].

¹⁹ Es kann sich um Eigenschaftssätze eines IFC Produktmodells (IfcPropertySets) sowie um Eigenschaftssätze des AUTODESK-eigenen AEC Produktmodells (AEC Property Sets) handeln. Über den Importvorgang werden vorhandene IfcPropertySets in AUTODESK-konforme AEC Property Sets konvertiert.

3.3.1 Ereigniserkennung

Interaktionen des Benutzers innerhalb des virtuellen Entwurfsraums werden über Kontrollinstanzen (*Reactor* [19, 141]) detektiert. Das OMF stellt hierfür die Klassen `AcDbDatabaseReactor` und `AcEditorReactor` zur Verfügung. Über den `AcDbDatabaseReactor` können Änderungen an der Geometrie und Änderungen an Eigenschaftssätzen eines Objektes erkannt werden. Bei Benutzermodifikationen werden über den Reactor spezifische Nachrichten ausgegeben. Die Nachricht `objectModified` beschreibt eine allgemeine Änderung an einem Objekt, `propertySetModified` beschreibt Änderungen an den Eigenschaftssätzen eines Objektes, und somit auch Änderungen an den Randbedingungen für den Simulationsprozess. Informationen über das Hinzufügen und Entfernen eines Objektes erfolgen über die Nachrichten `objectAppended` und `objectErased`. Der `AcEditorReactor` detektiert über das ADT ausgeführte Befehle sowie deren Status. Dadurch kann eine parallele Initiierung automatischer Funktionen wie beispielsweise das Pre-Processing verhindert werden. Der synchrone Ablauf von Ereignissen kann unter dem ADT zur Blockierung oder gar zum Absturz des Programms führen. Über die Einführung von Ereignisfiltern können die Nachrichten eines Reactors präzisiert werden. Eine Auswahl wichtiger Ereigniserkennungen ist in der Abbildung 3.8 dargestellt.

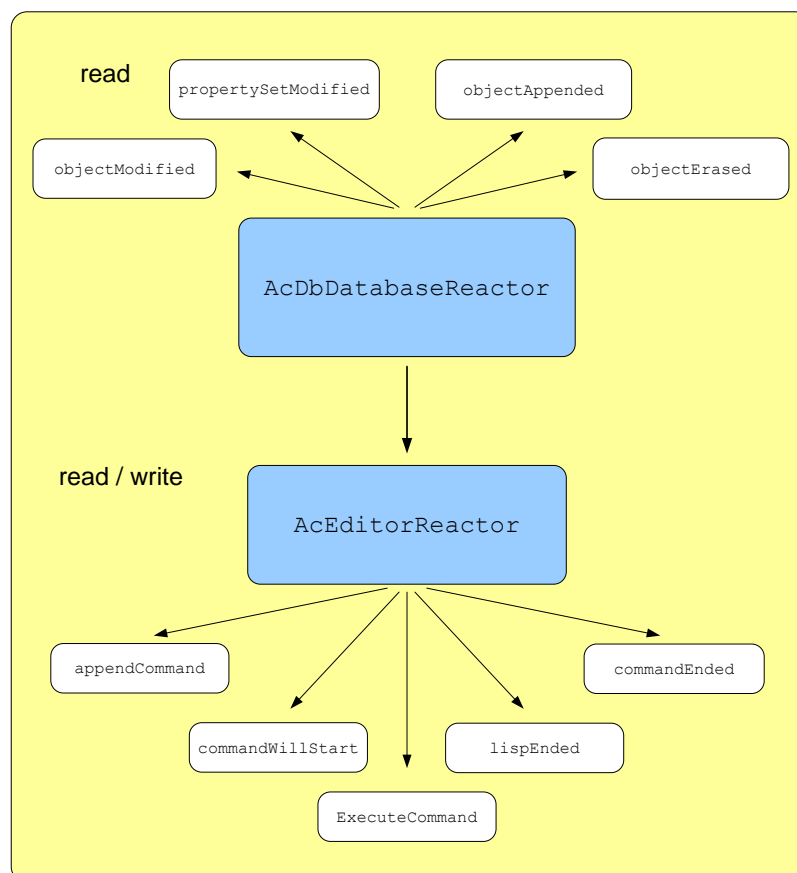


Abbildung 3.8: Reaktoren dienen der Detektion und Spezifizierung von Modifikationen des Benutzers innerhalb des ADT.

Ein Reactor kann im Sinne der Informatik als *Observer* [81] verstanden werden. Dieser versendet bei Eintritt eines bestimmten Ereignisses Mitteilungen an abhängige Instanzen. Die Mehrzahl der automatisierten Funktionen des Modellierers wird über Reactor-Benachrichtigungen initiiert. Hierzu zählen Korrekturfunktionen, Import- und Konvertierungsfunktionen, die Steuerung des HVAC-Agenten Frameworks und das Pre-Processing.

Datenbankstruktur des ADT. Zur Detektion von Modifikationen an Objekten ist der *AcDbDatabaseReactor* mit der Zeichnungsdatenbank des ADT verknüpft. Die Objekte einer Zeichnung werden bei ihrer Generierung über AutoCAD oder das ADT in einer Datenbank hinterlegt und entsprechenden Basisdatenbank-Objekten wie beispielsweise *Entities*, *Symbol Tables* und *Dictionaries* zugeordnet. *Entities* beschreiben Objekte zur grafischen Darstellung wie Linien, Kreise, Polygone und Text. Zusätzlich werden Datenbankeinträge von nicht sichtbaren Objekten, beispielsweise von Zeichnungsebenen, Text-Stilen und Dimensions-Stilen generiert. In einer Zeichnung gespeicherte Objekte werden mit dem in Abschnitt 3.2.1 beschriebenen Klassennamen-Präfix *AcDb-* deklariert. *Symbol Tables* und *Dictionaries* stellen Container für diese Datenbankobjekte dar. Genannt seien hier der *Block Table* für *Entities* oder der *Layer Table* für *Layer*. Die Struktur der AutoCAD/ADT Datenbank ist auszugsweise in der Abbildung 3.9 dargestellt [19, 141].

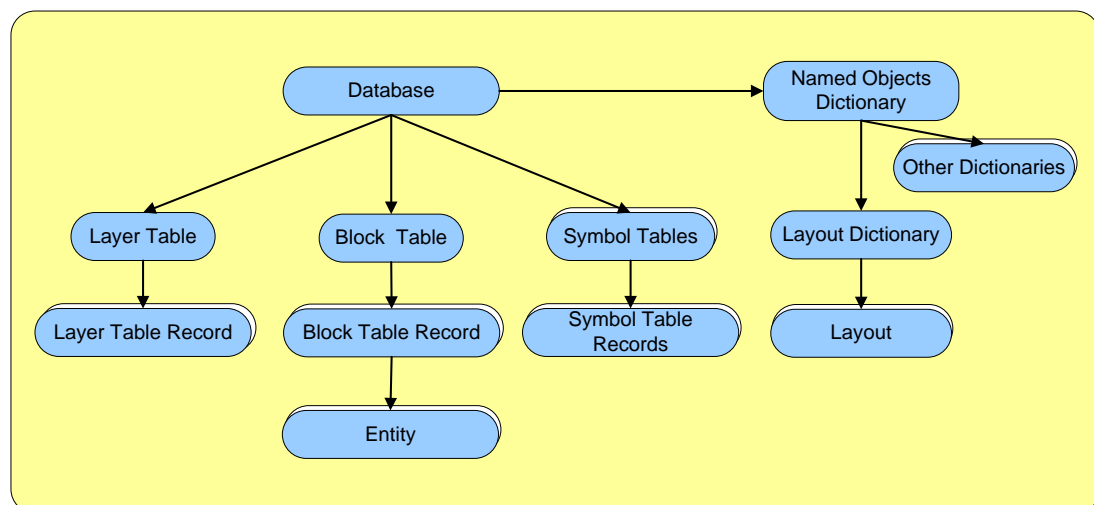


Abbildung 3.9: Struktur der Zeichnungsdatenbank von AutoCAD (Auszug)

3.3.2 Automatische Korrekturfunktionen

Die Funktionen des Software-Prototyps zur automatischen Fehlerdetektion und Korrektur werden über die Nutzung des *Reactors* sowie den im ADT enthaltenen *Automatic Property Definitions* [17] realisiert.

Die Position eines Objektes wird bei einer Benutzerinteraktion über Raumkoordinaten ausgelesen, analysiert und mit den Grenzkordinaten des Berechnungsgebietes verglichen. Bei Überschreitung wird eine Fehlermeldung ausgegeben und das Objekt wieder an eine erlaubte Position verschoben. Das Berechnungsgebiet kann automatisch oder manuell vom Benutzer über ein quaderförmiges Zeichenobjekt definiert werden. Bei einer

automatisierten Definition werden die geometrischen Grenzen der Konstruktionszeichnung analysiert und ihre maximale Ausdehnung als Simulationsraum festgelegt. Benutzereingaben zur Modifikation von Randbedingungen bei AEC Objekten, wie in der Abbildung 3.10 dargestellt, werden über Funktionen überprüft, welche vom ADT zur Verfügung gestellt werden.

The screenshot shows a software interface for defining properties of a 'Wand' (Wall) object. The 'EIGENSCHAFTSSÄTZE' (Property Sets) section is expanded, showing two sub-sections: 'limiting values' and 'simulation'. The 'limiting values' section contains fields for P_Max, P_Min, T_Max, T_Min, and u_max. The 'simulation' section contains fields for H, H_Soll, noslip, P, P_Soll, RAD_alpha, RAD_rho, RAD_rho_Soll, RAD_tau, RAD_tau_Soll, string_typ, T, T_Soll, type, u_x, u_x_Soll, u_y, u_y_Soll, u_z, and u_z_Soll. The 'Soll' (target) values are highlighted in grey, indicating they are write-protected. The 'EIGENSCHAFTEN' (Properties) tab is visible on the right side of the dialog.

Abbildung 3.10: Darstellung der Automatic Property Definitions. Über die Hinterlegung schreibgeschützter Grenzwerte (grau markierte Felder) kann die Kontrolle und automatische Korrektur von Benutzereingaben erfolgen.

Die Datenfelder eines Eigenschaftssatzes sind für Simulationsrandbedingungen über *Automatic Property Definition* deklariert und mit einem Eigenschaftssatz, welcher Grenzwerte (Limiting values) der Simulation beinhaltet, verknüpft. Die Eingabewerte (Soll-Werte) werden mit den erlaubten Minimal- und Maximalwerten verglichen und bei Überschreitung automatisch neu gesetzt. Die Eigenschaftssätze für Grenzwerte von Randbedingungen werden über eine *XML-Datenbank* [27, 218] vorgehalten und sind objekt- oder stilspezifisch definiert.

3.3.3 Pre-Processing

Die Aufbereitung der Berechnungsdaten über den virtuellen Entwurfsraum teilt sich in Funktionen der ereignisbasierten Erkennung, der Facettierung des Geometriemodells und in den Datenversand des Berechnungsmodells auf.

3.3.3.1 Ereignisbasierte Protokolle für den Sendeprozess

Ein Faktor zur Beschreibung der Effizienz eines Computational Steering Systems ist die Antwortzeit zwischen der Modifikation des Modells und der Visualisierung der Ergebnisse. Die im Rahmen dieser Arbeit durchgeführten Bemühungen zur Optimierung der Leistungsfähigkeit des Systems beschränkten sich hierbei nicht auf die Leistungsfähigkeit des Simulationskerns. Entscheidenden Einfluss hat auch der Einsatz effizienter Techniken für die Kommunikation zwischen den Modulen. Neben einer breitbandigen Datenleitung für einen schnellen Transfer des Berechnungsmodells zwischen Modellierungs-/Visualisierungsrechner und PC-Cluster dienen effiziente Transferprotokolle dem gefilterten Versand der Daten. Für das Pre-Processing wurden Sendeprotokolle entwickelt, welche über eine Analyse der Modifikationen am Berechnungsmodell fallspezifisch den effizientesten Datentransfer wählen. Im Sinne des Computational Steering Ansatzes ist eine exakte Fallunterscheidung für den Berechnungsstatus des Simulationskerns notwendig. Beispielsweise erfolgt bei lokalen Änderungen an Geometrie oder Randbedingungen kein Neustart der Simulation. Der Simulationslauf bleibt bestehen und die Randbedingungen des Strömungsgebietes werden aktualisiert. Die implementierten Sendeprotokolle unterscheiden vier spezifische Fälle:

1. Änderung der globalen Randbedingungen für die Berechnung, Neustart der Simulation, initialer Start der Simulation
2. Änderung der vorhandenen Geometrie des Modells
3. Hinzufügen eines neuen Geometrieobjektes
4. Änderung an objektbezogenen Randbedingungen für die Berechnung

Bei geometrischen Änderungen von Teilbereichen des Modells erfolgt eine globale Neuerstellung der Gesamtgeometrie. Dies erfolgt aufgrund von Einschränkungen des als Pre-Processor eingesetzten *Oktalbaum-Generator*²⁰(siehe Abschnitt 6.6 und [219]). Eine effizientere Fallunterscheidung kann über die Einführung einer *lokalen Diskretisierung* des Berechnungsgitters erfolgen.

²⁰ Der eingesetzte Oktalbaum-Generator erstellt bei einer Geometrieänderung ohne weitere Fallunterscheidung einen neuen Oktalbaum. Über eine lokale Diskretisierung ist es möglich nur den betroffenen Oktant, welcher das Gebiet der Geometrieänderungen umschließt, neu zu generieren. Weitere Oktanten des Oktalbaums bleiben erhalten. Dieser Ansatz führt zu einer Effizienzsteigerung hinsichtlich Speicherverbrauch, Datentransfer und Diskretisierungszeit.

Fall 1: Änderung der globalen Randbedingungen für die Berechnung, Neustart der Simulation, initialer Start der Simulation. Die Ausführung des Protokolls erfolgt über eine automatisierte Ereigniskette: die globalen und objektbezogenen Randbedingungen, welche über den ADT als Eigenschaftssätze (AEC Property Sets) vorliegen, werden mit der Methode `readPropSets` ausgelesen und in einer Liste zusammengefasst. Die Methode der Klasse `ADTPropSetUtilities` durchläuft hierfür die Datenbank des ADT und liest alle objektbezogenen Eigenschaftssätze der dort hinterlegten Elemente ein. Werden Eigenschaftssätze als Randbedingungen identifiziert, erfolgt die Extraktion von Objektinformationen und die Übertragung in die Sendeliste des Protokolls. Der Ablauf des ersten Szenarios ist in der Abbildung 3.11 dargestellt.

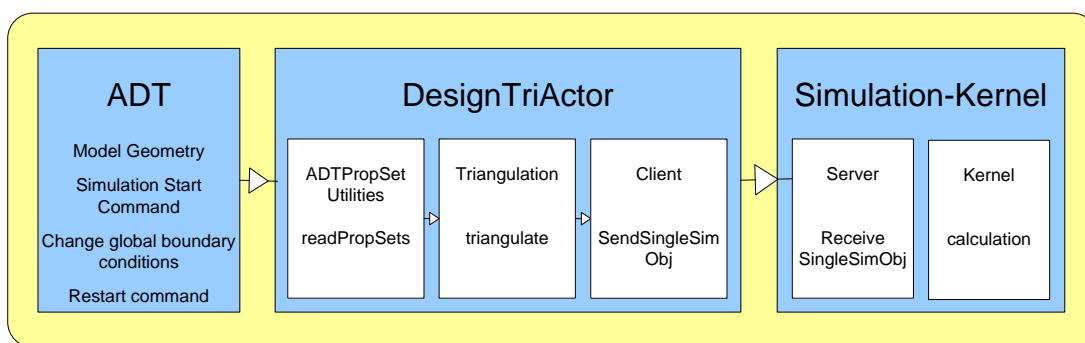


Abbildung 3.11: Schema zum Ablauf des Sendeprotokolls über das Szenario I

Nach Abschluss des Prozesses durchläuft die Methode `triangulate` über die ADT Datenbank eine Liste aller sichtbaren geometrischen Objekte. Jedes Objekt wird in ein Facettenmodell konvertiert und die geometrischen Informationen einer temporären Liste für den Sendeprozess hinzugefügt. Für nähere Informationen wird auf Abschnitt 3.3.3.2 sowie auf Abschnitt 6.5 verwiesen.

Objektgeometrie und objektbezogene Simulationsrandbedingungen werden getrennt vorgehalten. Ihre eindeutige Identifizierung erfolgt über eine vom ADT vergebene Object-ID (`AcDbHandle`). Dieser Ansatz erhöht die Effizienz der Kommunikation durch die getrennte Datenextraktion der ereignisbasierten Protokolle. Bei Modifikationen von Randbedingungen oder der Geometrie der Objekte ist hierdurch die Generierung beider Listen nicht erforderlich.

Abschließend werden die Listen an den Client der Socket-Kommunikation (vgl. Beschreibung in Abschnitt 3.3.3.2) zur Durchführung des Datentransfers an den Kommunikationsserver des Simulationskerns übergeben.

Die Initiierung des beschriebenen ersten Protokolls erfolgt ebenfalls bei der Einleitung eines Neustarts sowie bei der Modifikation von globalen Randbedingungen.

Fall 2: Änderung der existenten Geometrie des Modells. Die Ereigniskette des zweiten Szenarios wird bei Verschiebung, Dimensionsänderung oder Löschung geometrischer Objekte aktiviert. Die Ereignisdetektion erfolgt über den `AcDbDatabaseReactor`, die weitere Fallspezifizierung über Filterfunktionen des Reactors. Die dabei aktivierten Prozesse entsprechen in Teilbereichen denen des ersten Szenarios. Der sequenzielle Ablauf wird in der Abbildung 3.12 dargestellt.

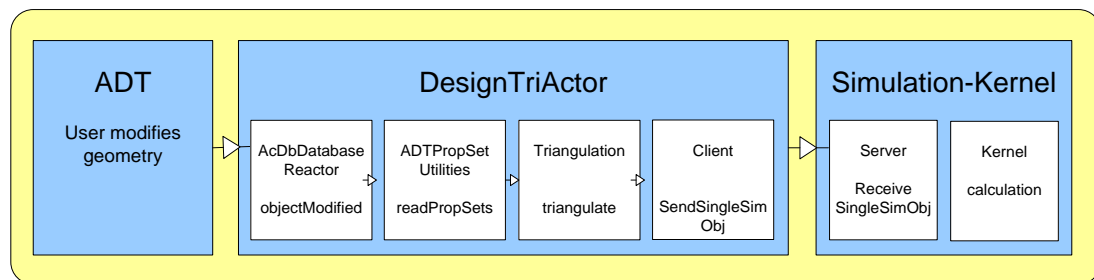


Abbildung 3.12: Schema zum Ablauf des Sendeprotokolls über das Szenario II

Das zweite Szenario unterscheidet sich gegenüber dem ersten Szenario in der Übermittlung von Daten. Es erfolgt innerhalb der Computational Steering Umgebung kein Neustart des Simulationslaufs und globale sowie objektbezogene Randbedingungen bleiben erhalten. Somit reduziert sich der Datenversand auf die geometrischen Informationen mit einer objektbezogenen ID. Die Identifizierung der Geometrieobjekte und die Zuordnung zu den aktuellen Randbedingungen des CFD-Kerns erfolgt über das `AcDbHandle`, welches über die gesamte Laufzeit der Simulation seine Gültigkeit beibehält. Bei einem Neustart des ADT erfolgt die Neuvergabe des objektbezogenen `AcDbHandle` durch den Neuaufbau der internen Datenbank. Eine Identifikation der Objekte über die von den IFC zur Verfügung gestellte und in den Eigenschaftssätzen der Objekte vorgehaltene `GUID` [167] erfolgt nicht. Der Grund hierfür liegt in der Größe des Datentyps einer `GUID`, welcher das Übertragungsvolumen des Datentransfers unnötig erhöhen würde. Der über die `GUID` erhaltene Mehrgewinn einer eindeutigen Identifikation eines Bauteilelementes über die Lebenszeit des Produktmodells erscheint im Rahmen des Computational Steering Simulationslaufs nicht sinnvoll.

Fall 3: Hinzufügen eines Geometrieobjektes. Das Hinzufügen eines geometrischen Objektes durch den Benutzer wird über den `AcDbDatabaseReactor` detektiert. Bei Generierung wird das neue Objekt mit objektbezogenen Randbedingungen für den Simulationsprozess aufbereitet. Der `AcEditorReactor` aktiviert den Schreibzugriff auf das entsprechende Objekt, welches nach der Identifikation des Objekttyps über die Methode `attachPropSet` der Klasse `ADTPropSetUtilities` ein spezifisches Property Set mit den objektbezogenen Randbedingungen erhält. Kann keine Identifikation erfolgen, werden Standardrandbedingungen gesetzt (Abbildung 3.13).

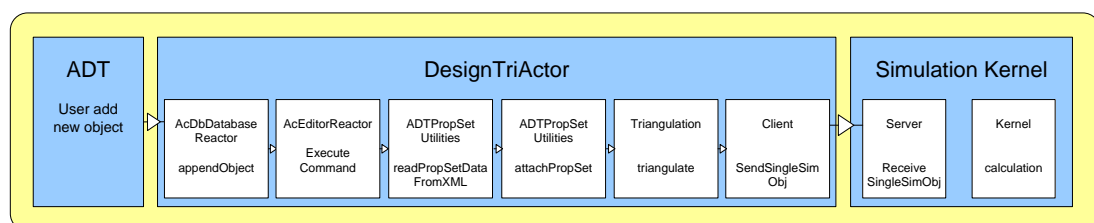


Abbildung 3.13: Schema zum Ablauf des Sendeprotokolls über das Szenario III

Der nachfolgende Ablauf des dritten Szenarios erfolgt analog zu dem Protokoll des ersten Szenarios. Es werden die vollständigen Informationen der Geometrie und der Randbedingungen an den Rechenkern übertragen.

Fall 4: Änderung an objektbezogenen Randbedingungen für die Berechnung. Änderungen der Eigenschaftssätze von Objekten durch den Benutzer werden über den `AcDbDatabaseReactor` detektiert und ausgewertet. Bei Modifikationen von Eigenschaftssätzen für Simulationsrandbedingungen wird der aktualisierte Datensatz über die `ADTPropSetUtilities` ausgelesen und mit dem `AcDbHandle` des Objektes an den Client für den Sendeprozess übergeben (Abbildung 3.14).

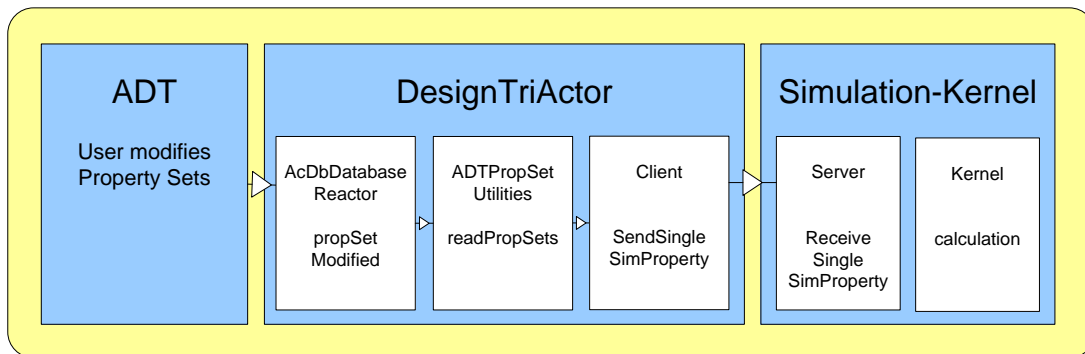


Abbildung 3.14: Schema zum Ablauf des Sendeprotokolls über das Szenario IV

Da kein Eingriff in die Geometrie des Modells erfolgt ist ein Versand von geometrischen Informationen nicht notwendig. Es wird nur der Eigenschaftssatz des betroffenen Elements zur Aktualisierung der Liste der Randbedingungen an den Rechenkern übertragen.

3.3.3.2 Sendeprozess

Eingesetzte Techniken. Der innerhalb des ADT implementierte Client für den Kommunikationstransfer und der Empfangs-Server des Rechenkerns nutzen eine *TCP Socket* Verbindung. Das *Transmission Control Protocol (TCP)* [48, 100] dient als Basis für mehrere Anwendungsschichten von Internetprotokollen wie *FTP (File Transfer Protocol)* [159] oder *HTTP (Hypertext Transfer Protocol)* [160]. Es kann ebenfalls für die Entwicklung eigener Protokolle genutzt werden. Das TCP ist datenstromorientiert und benötigt für den Datenaustausch eine Client- sowie eine Serverapplikation.

Die für den Kommunikationsprozess des Prototyps implementierte TCP Verbindung basiert auf der über das *Qt Framework* von TROLLTECH [209] zur Verfügung gestellte Klasse `QSocket`. Über ihre Methode `QDataStream` bietet die Klasse einen effizienten block-orientierten Datentransfer [208]. Die auf C++ aufbauende *Qt Klassenbibliothek* ist plattformunabhängig. Somit ist ein Einsatz über das ADT als Client-Applikation unter *MICROSOFT Windows* sowie der Einsatz über den Rechenkern als Server-Applikation unter *Debian Linux* [193] möglich.

Kommunikationsprotokoll. Das im Rahmen dieser Arbeit entwickelte Kommunikationsprotokoll dient dem Transfer von Randbedingungen und geometrischen Informationen des virtuellen Entwurfsraums an den CFD-Kern und ist an die in Abschnitt 3.3.3.1 beschriebenen Ereignisprotokolle gekoppelt. Es werden drei Fälle unterschieden:

- *Neugenerierung:* Der Transfer erfolgt bei Eintritt des ersten Szenarios des Ereignisprotokolls und beinhaltet einen vollständigen Versand aller geometrischen Objekte sowie aller Randbedingungen des Modells zur Neugenerierung des Rechengitters.
- *Geometrieänderung:* Die Modifikation geometrischer Objekte und das Hinzufügen oder Löschen von Objekten erfolgt über die Szenarien II und III des Ereignisprotokolls. Beide Fälle resultieren in einem Versand der geometrischen Informationen aller Objekte sowie der Objekt IDs über das Sendeprotokoll, da über den Oktalbaum-Generator ein neues Gitter generiert wird. Die Randbedingungen der Objekte bleiben bei der Aktualisierung erhalten und werden nicht versendet. Bei dem Hinzufügen eines Objektes werden die Randbedingungen dieses Elements in den Sendeprozess aufgenommen.
- *Änderung der Randbedingungen:* Bei Eintreten von Szenario IV des Ereignisprotokolls wird der Eigenschaftssatz der Randbedingungen des modifizierten Elementes mit der dazugehörigen Objekt-ID versendet. Über den Rechenkern erfolgt eine Aktualisierung der Liste der Objekteigenschaften. Der Versand geometrischer Informationen zur Generierung eines neuen Rechengitters ist nicht erforderlich.

Die Listen des Sendeprozesses für die geometrischen Informationen sowie die Randbedingungen aller Objekte des Modells werden für einen einmaligen Versand temporär generiert. Globale Parameter für die Simulation und für Objektinformationen, bestehend aus dem Eigenschaftssatz für Randbedingungen, der ID und einer Liste mit Koordinaten der Dreiecksfacetten werden über das Sendeprotokoll in Datenblöcke unterteilt. Kopf und Rumpf eines Blocks enthalten Zusatzinformationen über den Status des Ereignisprotokolls sowie über die Blockgröße. Die Daten für den Kommunikationsprozess werden blockweise an den Server übertragen. Nach dem Versand der Informationen an den Server wird die TCP-Verbindung unterbrochen. Dieses dient der Freigabe des ADT für andere Prozesse, da der Kommunikationsprozess für den Zeitraum der Datenverbindung die Applikation blockiert. Der Empfangs-Server des Rechenkerns wird über eine *Thread-Programmierung* realisiert und ist permanent aktiv. Der Server überwacht einen definierten Port auf eingehende Verbindungsanfragen des Clients. Für eine detaillierte Beschreibung zum Aufbau des Kommunikationsprotokolls wird auf [219] verwiesen.

Das hier vorgestellte Protokoll zeigt bei Benchmark-Tests eine hohe Effizienz. So wurde für die Triangulation, die Listenerstellung und den Datenversand eines Modells des UNIQA-Towers [211] eine Gesamtzeit von unter zwei Sekunden gemessen. Das Modell weist eine komplexe Geometrie mit über 35.000 Dreiecksfacetten auf. Die für den Benchmark-Test eingesetzte Hardware ist in Appendix A beschrieben.

3.3.4 Automatische Erstellung von Bauplänen

Der *LayoutManager* [29] wurde über die Programmierschnittstelle OMF entwickelt und erweitert den Modellierer um Funktionen zur automatischen Erstellung von spezifischen Bauplänen für die am Konstruktionsprozess beteiligten Fachplaner. Hierbei wird der objektorientierte Ansatz des Produktmodells aufgegriffen, um Bauteilen zusätzliche fachplanerspezifische Filterattribute zuzuweisen [15]. Durch die Zuweisung fallspezifischer geometrischer Darstellungen an ein Objekt des AEC Produktmodells können anforderungsabhängige Darstellungen gewählt werden. Die Darstellung eines Objekts ist beispielsweise als zweidimensionale Geometrie für einen Bauplan, als dreidimensionale Isometrie, oder als fachplanerspezifisches Symbol möglich.

Das Bauplanmodul des Prototyps filtert Informationen zur geometrischen Bauteilzuordnung und weist die Darstellungen eines Objektes abhängig von ihren Geometrieattributen spezifischen Bauplänen zu. Dies erfolgt im Rahmen dieser Arbeit exemplarisch für die in der Abbildung 3.15 dargestellten Bereiche der Hausleittechnik (Leitungen, Klimaanlagelemente), der Architektur (Mauerwerk, Stürze, Durchbrüche) und der Innenarchitektur (Trennwände, Möblierung).

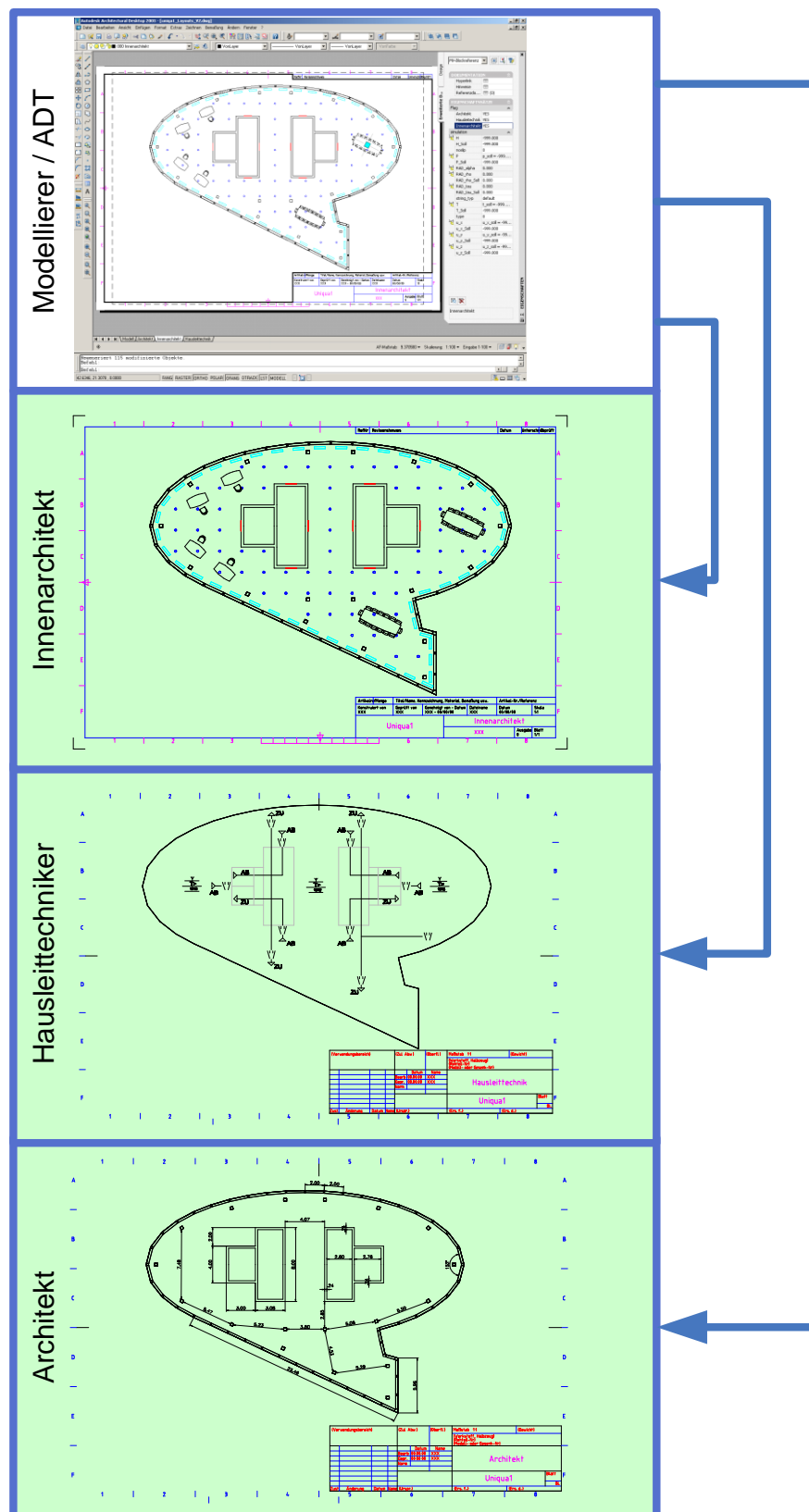


Abbildung 3.15: Automatische Erstellung fachspezifischer Baupläne

Die Weiterverarbeitung der Pläne kann neben der Ausgabe auf einen Plotter auch über verschiedene digitale Datenformate erfolgen. Die Abbildung 3.16 stellt den schematischen Aufbau des LayoutManagers in UML-Notation dar.

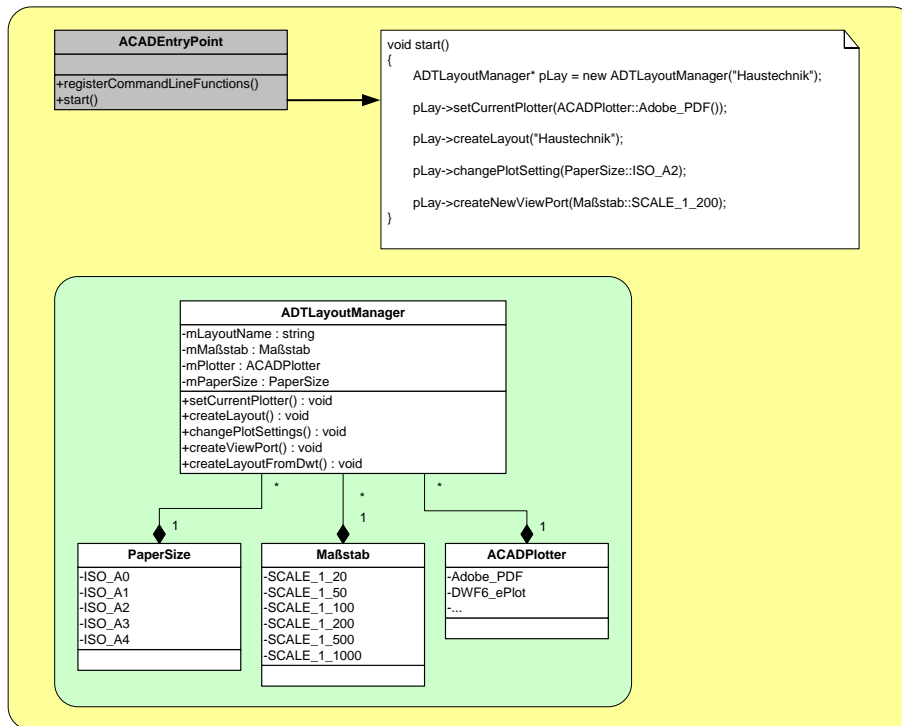


Abbildung 3.16: Aufbau des LayoutManagers

Neben Funktionen zur fachspezifischen Filterung geometrischer Attribute werden Eigenschaften des Bauplanes wie Maßstab, Ausgabemedium und Ausgabeformat berücksichtigt. Die Baupläne der Fachplaner werden bei der Generierung einer eigenen Zeichenebene zugeordnet.

3.4 Implementierung der Software-Module

Die Abbildung 3.17 vermittelt einen Überblick über die Implementierung der entwickelten Softwaremodule für den virtuellen Entwurfsraum, welche in Abschnitt 3.3 erläutert wurden. Die Abbildung zeigt das Zusammenspiel einzelner Komponenten mit dem ADT und die Interaktion der Komponenten untereinander.

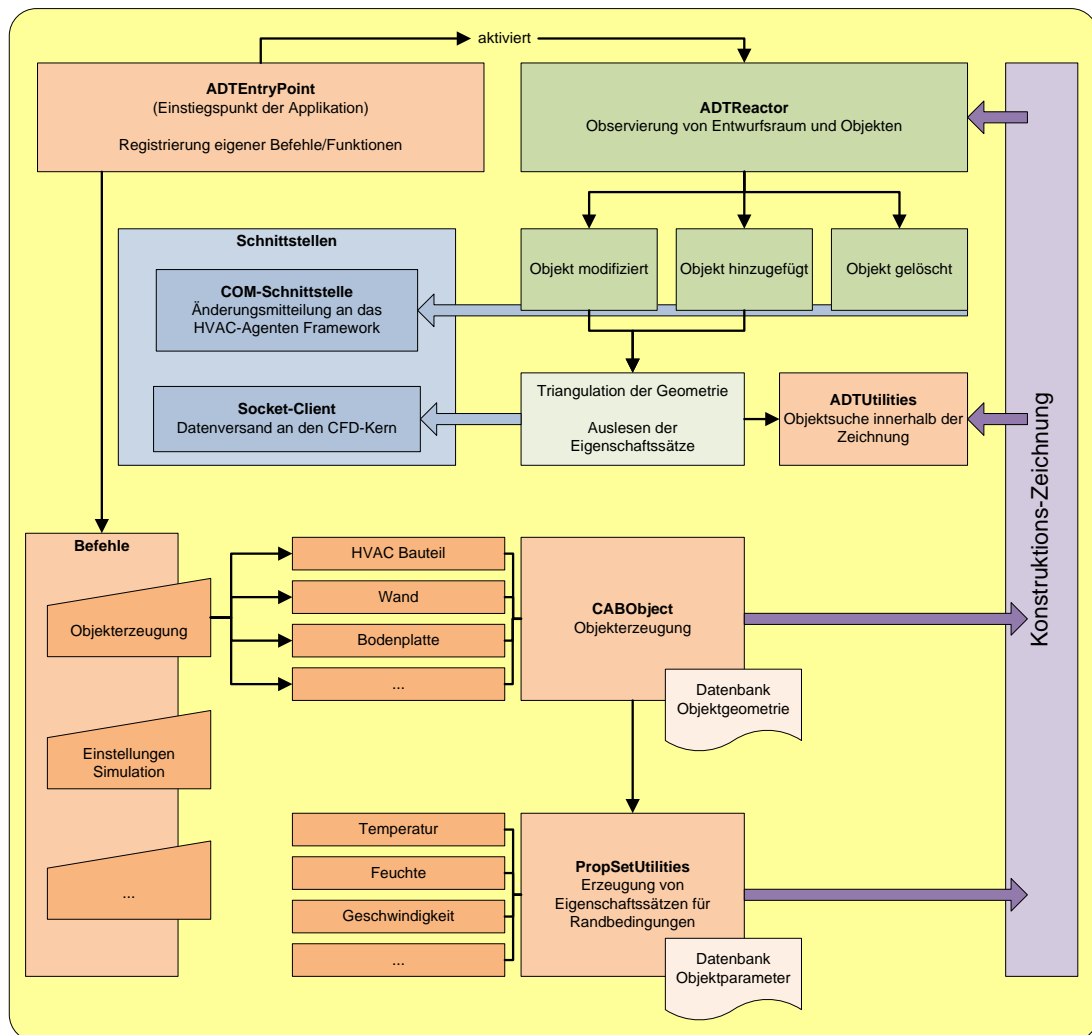


Abbildung 3.17: Schematische Darstellung der implementierten Softwaremodule in das ADT

3.5 Schlussfolgerung und Ausblick

Möglichkeiten des ADT/OMF. Die Ergebnisse der Arbeiten mit den AUTODESK Programmierschnittstellen OMF und ObjectARX zeigen, dass die Entwicklung eigener, Produktmodell unterstützender Applikationen durch den hohen Funktionsumfang dieser Schnittstellen realisierbar ist. Das OMF bietet die Möglichkeit eines Zugriffs und einer Manipulation von Elementen des Produktmodells und erlaubt über seine Klassenbibliothek die effiziente Erstellung komplexer und automatisierter Funktionen.

Durch die enge Beziehung zwischen dem AUTODESK AEC Produktmodell und dem IFC Modell ist die Möglichkeit einer datenkonformen Erweiterung von IFC Objekten um simulationsspezifische Attribute durch Nutzung der IFC und AEC Eigenschaftssätze (Property Sets) gegeben. Durch die direkte Unterstützung bauteilspezifischer Informationen innerhalb des ADT kann eine optimale Integration eigener Funktionen und Bauteilelemente in die vorhandene Oberfläche der Applikation erfolgen. Diese Möglichkeit erhöht die Benutzerfreundlichkeit der Applikation und erübrigt die Einführung eigener grafischer Oberflächen. Die für die Realisierung entscheidenden Funktionen des ADT stellen hierbei die Werkzeug- und Eigenschaftspaletten, der Stil-Manager, die automatischen Eigenschaftsdefinitionen und die Ankerelemente dar.

Schwächen des ADT/OMF. Die Entwicklung hochintegrierter Applikationen über das ADT kann nur über die OMF-Schnittstelle erfolgen. Ein Zugriff auf Bauteile und bauteilspezifische Informationen des Produktmodells über die ObjectARX Schnittstelle ist nicht möglich. Die Einarbeitungszeit in die Funktionsweisen des Object Modeling Frameworks kann gegenüber der Einarbeitungszeit in ObjectARX als hoch eingestuft werden. Dies ist zum einen mit der Komplexität der Klassenbibliothek zu begründen, zum anderen ist die Dokumentation für das OMF unzureichend und veraltet²¹. Klassen und Methoden sind teilweise nicht dokumentiert und müssen in einem langwierigen iterativen Prozess mit den AUTODESK Entwicklern erörtert werden.

Der ADT zeigt Schwächen bei der Entwicklung automatisierter Prozesse. Dies ist auf die generelle Softwarearchitektur des Programms zurück zu führen. So unterstützt das ADT nicht die gleichzeitige Nutzung parallel laufender Prozesse (*Threads*), welche bei der Realisierung einer interaktiven Softwareapplikation für automatisierte Abläufe von Hintergrundprozessen notwendig sind. Eine Notlösung findet sich in einem sequenziellen Ablauf von Teilprozessen des Programms. Durch temporäres Blockieren einzelner Instanzen ist eine Automatisierung gewährleistet, dies führt aber zu Einschränkungen hinsichtlich eines flüssigen Programmablaufes des Prototyps. Die Funktionen zur HVAC Optimierung wurden aufgrund der beschriebenen Einschränkungen vollständig ausgelagert und sind als eigenständiges Programm realisiert.

Der ADT zeigt in vielen Fällen Inkompatibilitäten bei der Implementierung von Fremdapplikationen. Zur Erreichung eines stabilen Systems musste eine sorgfältige Validierung von Klassen erfolgen, welche für Funktionen wie die Socket-Kommunikation oder den XML-Datenimport benötigt wurden.

²¹ Die von AUTODESK zur Verfügung gestellte Dokumentation zur Entwicklung von OMF-Applikationen ist die *OMF Developer's Guide 3.3*. Die Inhalte des Dokumentes beziehen sich auf den ADT 3.3. Zum Zeitpunkt dieser Arbeit sind bereits fünf Nachfolgerversionen des Architectural Desktop erschienen, ohne dass eine Aktualisierung der Dokumentation erfolgte.

Kapitel 4

Das HVAC-Agenten Framework

4.1 Einleitung

Die Luftzuführung aktueller Klimatisierungsanlagen erfolgt nicht mit einer konstanten Temperatur oder Geschwindigkeit. Elektronische Systeme reagieren vielmehr auf die klimatischen Änderungen innerhalb eines Raumes und passen den Luftstrom permanent den dynamisch variierenden Umwelteinflüssen an. Zuständig für diese Abläufe sind *mess-, steuer- und regeltechnische Prozesse (MSR-Technik)*. Hierbei werden verschiedene Komponenten zur Messung (beispielsweise Temperaturfühler) und Steuerung (beispielsweise *DDC-Prozessoren*²²) eingesetzt, welche entsprechend ihrer bautechnischen Eigenschaften und implementierten Regelwerke (digitale oder analoge Steuerung) ein spezifisches Verhalten aufweisen. Durch die Kopplung einer Vielzahl dieser Komponenten entsteht ein intelligenter Verbund, dessen Verhalten aufgrund seiner Komplexität nur über eine numerische Simulation genau beschrieben werden kann.

Ein wichtiger Teilaspekt der hier vorgestellten Simulationsumgebung ist die realistische Emulation von Klimaanlagekomponenten, so genannter *HVAC-Systeme (Heating, Ventilation and Air Conditioning, [13, 177])*, welche innerhalb des Konstruktionsraumes positioniert werden und einen bedeutenden Einfluss auf die globale Raumluchtströmung nehmen.

Die komponentenspezifischen Regelwerke können eine hohe Komplexität aufweisen, da ihr Verhalten durch die heute gängige und kostengünstig herzustellende Digitaltechnik (*DDC-Technik; [176, 184]*) stark verzweigte Charakteristiken aufweisen kann. Hierbei sind gegenüber den analogen Techniken (beispielsweise Thermostate oder Steuerrelais) nicht nur einfache Funktionen (beispielsweise lineares oder exponentielles Verhalten) möglich. Es können über logische Verknüpfungen beliebig komplexe Regelungstechniken frei programmiert werden. Ein in der Praxis angewandtes Steuerverhalten basiert auf Regelwerken, welche die Randbedingungen für den Steuerprozess über *Behaglichkeitsfelder* (siehe hierzu Abschnitt 4.6.3 und [176, 184]) definieren. Die optimale Raumlucht wird hierbei unter Berücksichtigung von Temperatur, Geschwindigkeit und Feuchtigkeit für eine spezifische Raumnutzung (beispielsweise Büro, Maschinensaal, Konzertsaal) zur Reduzierung der Energiekosten berechnet. Die hierbei hinterlegten Regelwerke weisen ihrerseits keinen starren Charakter auf. Sie können über beliebig geartete Parameter den jeweiligen Anforderungen angepasst werden. Dies Beispiel zeigt, dass neben der geometrischen Anordnung der Komponenten noch weitere Randbedingungen für den

²² DDC steht für **D**irect **D**igital **C**ontrol und beschreibt mikroprozessor-gesteuerte Steuerungs- und Regelungs-Systeme im Bereich der Heizungs- und Klimatechnik.

Simulationsprozess zu berücksichtigen sind, die in ihrer Gesamtheit ein optimal agierendes System definieren. Für die Lösung dieser Probleme ist in der Praxis der Bereich der *Hausleittechnik* spezialisiert.

Das hier beschriebene Anwendungsfeld erfordert aufgrund seiner vielschichtigen Verknüpfungsmöglichkeiten sowie der komponentenspezifischen Optimierung einzelner Parameter ein eigenes Simulations-Segment, welches als „Simulation innerhalb der Simulation“ beschrieben werden kann. Eine logische Konsequenz für die virtuelle Abbildung solcher Klimaanlagekomponenten mit integrierten Regelwerken ist die Einbindung von *Software-Agenten*, einem innovativen Ansatz der Informatik aus dem Bereich der *Künstlichen Intelligenz (KI)* (siehe hierzu [69, 72, 178, 217] und Abschnitt 4.4).

Die Implementierung dieser Ansätze in den Simulationsprozess soll nach einer Einführung in die Klimatechnik, die Regeltechnik und die Software-Agenten-Technologie innerhalb dieses Kapitels genauer erläutert werden.

4.2 Thermische Behaglichkeit

Für die Optimierung von Raumluftrömungen innerhalb eines Gebäudes oder Gebäudeabschnittes über eine numerische Simulation ist ein Referenzwert hinsichtlich der Qualität des Umgebungsklimas erforderlich. Diese Qualität kann über einen skalaren Wert, welcher das thermische Behaglichkeitsempfinden eines Menschen prognostiziert, wiedergegeben werden. Die optimale lokale thermische Behaglichkeit wird über den in der DIN EN ISO 7730 [106] definierten *PMV*²³- und *PPD*²⁴-Index interpretiert. Dieser Index bewertet unter anderem Lufttemperatur, Luftfeuchte, Luftbewegung und Wärmestrahlung unter Berücksichtigung des Bekleidungsgrades und der Art der Aktivität der Personen im Raum (siehe Abbildung 4.1).

²³ PMV = engl. „**P**redicted **M**ean **V**ote“ oder dt. „vorausgesagtes mittleres Votum“

²⁴ PPD = engl. „**P**redicted **P**ercentage of **D**issatisfied“ oder dt. „Vorausgesagter Prozentsatz an Unzufriedenen“

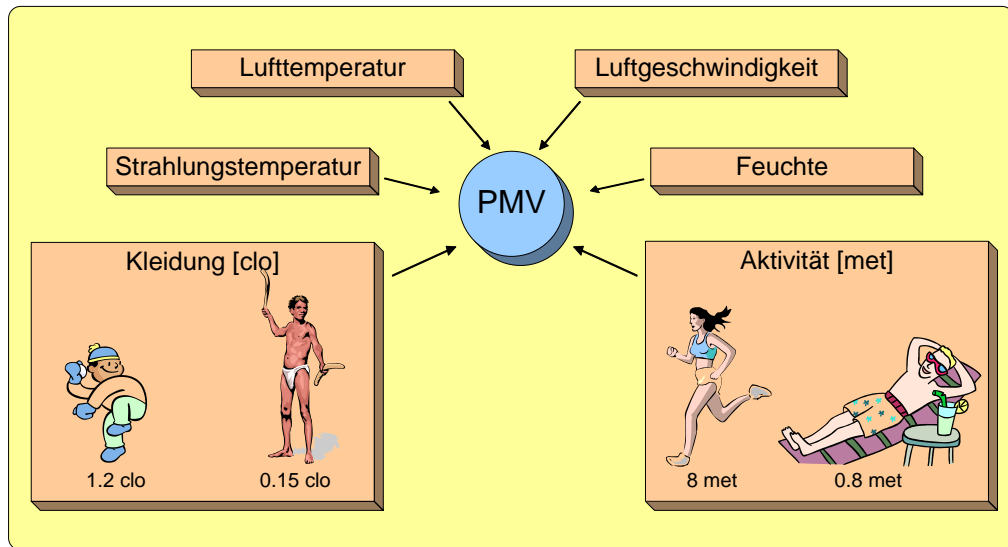


Abbildung 4.1: Der PMV-Index (Predicted Mean Vote) berücksichtigt mehrere klimarelevante Parameter zur Definition der thermischen Behaglichkeit eines Menschen.

Das menschliche Wärmeempfinden hängt vom thermischen Gleichgewicht des *Körpers als Ganzem* ab (*Wärmebilanz*). Nach einer Messung oder Schätzung der notwendigen Faktoren kann der PMV-Index über folgende Formel berechnet werden:

$$\begin{aligned}
 PMV = & (0,303 e^{-0,036 M} + 0,028) \\
 & \cdot [(M - W) - 3,05 \cdot 10^{-3} \{5733 - 6,99(M - W) - p_a\} \\
 & - 0,42 \{(M - W) - 58,15\} - 1,7 \cdot 10^{-5} M (5867 - p_a) \\
 & - 0,0014 M (34 - t_a) - 3,96 \\
 & \cdot 10^{-8} f_{cl} \{(t_{cl} + 273)^4 - (\bar{t}_r + 273)^4\} - f_{cl} h_c (t_{cl} - t_a)]
 \end{aligned}$$

Beinflussende Faktoren sind hierbei:

- Der Energieumsatz M , welcher in $\frac{W}{m^2}$ oder *met* angegeben wird. Eine entspannt sitzende Person hat beispielweise einen Energieumsatz von $46 \frac{W}{m^2}$ oder $0,8 \text{ met}$, bei einem Skifahrer liegt dieser Wert bei $405 \frac{W}{m^2}$ oder $7,0 \text{ met}$ [101].
- Die wirksame mechanische Leistung W , welche ebenfalls in $\frac{W}{m^2}$ oder *met* angegeben wird.
- Der dimensionslose Bekleidungsfaktor f_{cl} , welcher sich über folgende Formel berechnet:

$$f_{cl} = 1,00 + 1,290 I_{cl}$$

Hierbei geht die Bekleidungsisolations I_{cl} in die Berechnung mit ein, welche in $\frac{m^2 \cdot K}{W}$ oder *clo* angegeben wird. Jedem Kleidungsstück kann ein Bekleidungsisolationswert zugeordnet werden. So hat beispielsweise ein T-Shirt den Wert von $0,09 \text{ clo}$, ein Jackett den Wert von $0,35 \text{ clo}$ [101].

- Die Lufttemperatur t_a gemessen in $^{\circ}C$.
- Die mittlere Strahlungstemperatur \bar{t}_r gemessen in $^{\circ}C$.

- Der konvektive Wärmeübertragungskoeffizient h_c , welcher in $\frac{W}{m^2 \cdot K}$ gemessen wird. Er ergibt sich aus dem größeren Wert der beiden Formeln:

$$h_c = 12,1 \sqrt{v_{ar}} = 2,38(t_{cl} - t_a)^{0,25}$$
Wobei $v_{ar} \left[\frac{m}{s} \right]$ die relative Luftgeschwindigkeit und $t_{cl} [^{\circ}C]$ die Oberflächentemperatur der Bekleidung definieren.
- Der Wasserdampfpartialdruck p_a gemessen in Pa .

Die Ermittlung des PMV-Indexes erfolgt über die Beurteilung der körperlichen Tätigkeit durch die ISO 8996 (*Energieumsatz*, [112]), sowie der Bekleidung durch die ISO 9920 (*Wärmedurchlasswiderstand*, [110]). Die Parameter für Lufttemperatur, mittlere Strahlungstemperatur, relative Luftgeschwindigkeit und den partiellen Wasserdampfdruck sind über Messungen, Simulationen oder Schätzungen festzulegen (ISO 7726, [111]).

Der PMV-Index gibt einen Durchschnittswert zur Klimabeurteilung durch eine große Personengruppe über eine siebenstufige Klimabeurteilungsskala wieder. Die Skala reicht von dem Wert 0 (neutral) bis zu den Werten +3 (heiß) bzw. -3 (kalt). Der PMV-Index basiert auf dem Wärmegleichgewicht des menschlichen Körpers. Dieses wird erreicht, wenn die vom menschlichen Körper erzeugte Wärme der an die Umgebung abgegeben Wärme gleicht [106].

Neben einer Klimabeurteilung für die Gesamtheit einer Personengruppe streuen die Beurteilungen einzelner Personen um diesen Mittelwert. Über den *PPD-Index* (*Predicted Percentage of Dissatisfied*) [%] kann eine qualitative Voraussage über die Anzahl von Personen dargestellt werden, welche mit einem bestimmten Umgebungsklima unzufrieden sind. Der PPD-Index kann, ausgehend von PMV-Werten, über die in der Abbildung 4.2 dargestellte Formel berechnet werden.

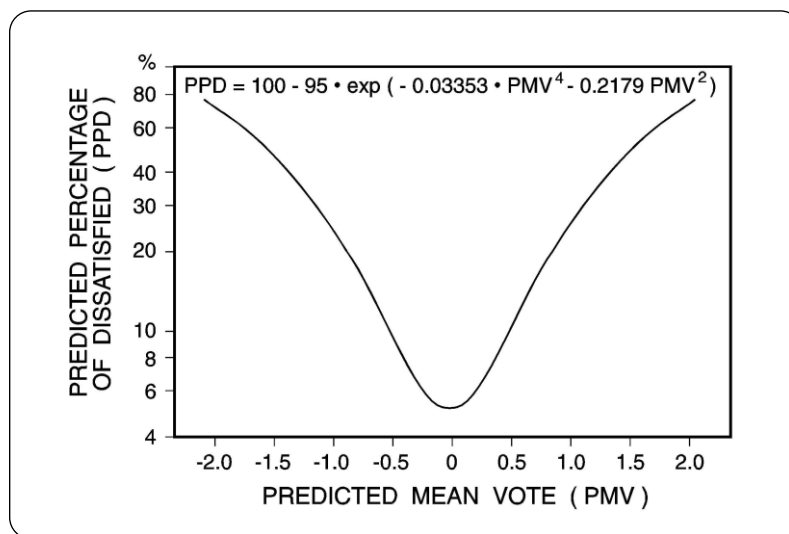


Abbildung 4.2: Der PPD-Index (*Predicted Percentage of Dissatisfied*) berechnet sich aus dem PMV-Index und beschreibt den prozentualen Anteil von Personen, die ein bestimmtes Umgebungsklima als „zu warm“ oder „zu kalt“ empfinden.

Lokale thermische Behaglichkeit. Neben dem PMV- und dem PPD-Index, welche das Behaglichkeitsempfinden als Ganzes definieren, sind *lokale thermische Einflüsse* zu beachten. Dazu zählen Faktoren wie *Zugluft*, *vertikale Temperaturunterschiede* (im Bereich zwischen Kopf und Fuß), *zur Raumtemperatur stark differierende Fußböden* und eine *hohe Asymmetrie der Strahlungstemperatur*, die zu einem unerwünschten Abkühlen oder Aufheizen eines Körperteils führen. Diese Faktoren haben ihren höchsten Einfluss auf das Behaglichkeitsempfinden bei der Ausführung von leichter, sitzender Tätigkeit. Das Vorgehen zur Berücksichtigung von Einflussfaktoren zur lokalen Behaglichkeit ist in der ISO 7730 aufgeführt. Unter anderem kann der Prozentsatz *DR* von Personen, welche auf Grund der Beeinträchtigung durch Zugluft unzufrieden sind, über folgende Formel berechnet werden [106]:

$$DR = (34 - t_a)(v_a - 0,05)^{0,62}(0,37 \cdot v_a \cdot Tu + 3,14)$$

Einfluss hat hierbei der lokale *Turbulenzgrad* *Tu* [%], ein Quotient aus der Standardabweichung der *lokalen Luftgeschwindigkeit* und der *lokalen mittleren Luftgeschwindigkeit*. Zur Feststellung von Turbulenzen im Raum führt die numerische Simulation gegenüber lokalen Messungen zu genaueren Ergebnissen.

Bewertung der thermischen Behaglichkeit. Zur Auslegung einer HVAC-Anlage werden in der Praxis die Behaglichkeitsbedingungen im Raum über Computersimulationen oder Messungen über längere Zeitspannen hinweg geprüft. Bei einer vollständigen Erfüllung der Behaglichkeitskriterien einschließlich Extremsituationen sind die notwendigen Wärme- bzw. Kühlleistungen der HVAC-Anlagen in der Regel sehr hoch. Unter Berücksichtigung von wirtschaftlichen und umwelttechnischen Faktoren gilt es als annehmbar, innerhalb begrenzter Zeitintervalle die festgelegten Bereiche des PMV-Indexes zu überschreiten.

4.3 Regelungstechnik

4.3.1 Regelungstechnik in der Praxis

Durch die Verbreitung von computergestützten und anderen digitalen Techniken kommt der *automatisierten Regelung* eine steigende Bedeutung zu. Die Vorteile für den praktischen Einsatz sind hoch, da neben den fallenden Produktionskosten für solche Komponenten die Messungs- und Regelungsergebnisse von Anlagen immer präziser werden. Somit können hohe Summen an Energiekosten eingespart werden. Verstärkt wird dieser Trend durch permanent ansteigende Energiepreise, erhöhte Auflagen durch den Umweltschutz und das zunehmende Komfortbedürfnis des Menschen.

Da das Aufgabengebiet für diesen Sektor sehr weit gefächert ist, hat sich in der Praxis der Bereich der *Gebäudeleittechnik* etabliert, welcher sich ausschließlich mit der Installation und Parametrisierung geregelter Anlagen beschäftigt. Ihre Aufgabengebiete sollen im Folgenden genauer spezifiziert werden [176, 199]:

- **Mess-, Steuer- und Regelungstechnik (MSR).** Dieser Bereich beschäftigt sich mit den allgemeinen Prinzipien und Techniken von Regelungsvorgängen, welcher in Abschnitt 4.3.2 genauer beschrieben wird. Begriffsdefinitionen sowie Aufbau und Funktion von Komponenten der *Leittechnik*, *Regelungstechnik* und *Steuertechnik* sind in Deutschland über DIN Normen standardisiert [56-61].
- **DDC-Technik.** Bereits seit 1979 sind in der Heizungs- und Klimatechnik *mikroprozessor-gesteuerte Steuerungs- und Regelungs-Systeme* im Einsatz (*DDC*). Bei der Steuerung dieser Anlagen wird die konventionelle *Relaistechnik* mit einer festen Verdrahtung durch eine *speicherprogrammierbare Steuerung* (*SPS-Steuerung*) ersetzt. Dadurch wird der elektrische Aufbau eines Schaltschranks mit seinen Leistungsschützen von den Relais entkoppelt. Da die logische Verknüpfung dieser Steuerung frei programmierbar ist, kann sie gegenüber der konventionellen Relastechnik ohne Neuverdrahtung jederzeit geändert werden. Bei der *DDC-Regelung* können Regelparameter frei programmiert werden. Moderne *DDC-Systeme* integrieren somit Regelung, Steuerung und Optimierung.
- **Gebäudeleittechnik.** Die Gebäudeleittechnik beschreibt einen Verbund von selbständigen, dezentralen, digitalen Regel- und Steuereinheiten (*DDC-Einheiten*) mit einer oder mehreren Zentraleinheiten. Dieser Verbund wird über einen manuell initiierten oder automatisierten Kommunikationsaustausch erreicht, welcher *intelligente* Datenübertragungseinrichtungen voraussetzt. Hierüber können anlagenspezifische Daten wie Messwerte, Ereignisse, Betriebs- und Alarmmeldungen, Stell- und Schaltsignale jederzeit abgefragt, gemeldet, protokolliert, gespeichert und zur weiteren Verarbeitung herangezogen werden. Beispiele hierfür sind Auswertungen von Funktionsabläufen, die Kontrolle von Optimierungsmaßnahmen und die Aufstellung von Verbrauchsberechnungen. Neben den Möglichkeiten zur Erstellung von Regel- und Steuerungsfunktionen in Verbindung mit Zeit- und Logikprogrammen stellt ein Gebäudeleittechniksystem eine Reihe von zusätzlichen Prozessen zur Verfügung. Diese ermöglichen die Einleitung von kontrollierten Maßnahmen zur Optimierung von Betriebsabläufen.

4.3.2 Einführung in die Regelungstechnik

4.3.2.1 Begriffe und Benennung

Über den ARBEITSAUSSCHUSS REGELUNGSTECHNIK wurden erstmalig im Mai 1968 die wichtigsten Begriffe der *Regelungs- und Steuerungstechnik* in den Normen der Reihe DIN 19226 [57-61] festgelegt:

Hierbei wird die *Steuerung* als Vorgang definiert, bei dem ein oder mehrere Eingangsgrößen andere Größen als Ausgangsgrößen aufgrund der Gesetzmäßigkeiten des Systems beeinflussen. Kennzeichen für die *Steuerung* ist der *offene Wirkungsablauf* über das einzelne Übertragungsglied oder die *Steuerkette*.

Die *Regelung* wird als ein Vorgang definiert, bei dem eine Größe und eine zu regelnde Größe (*Regelgröße*) kontinuierlich erfasst und fortlaufend mit einer weiteren Größe (*Führungsgröße*) verglichen werden. Abhängig von dem Ergebnis im Sinne einer Angleichung an die Führungsgröße wird nun die Regelgröße beeinflusst. Der beschriebene Wirkungsablauf findet in einem *geschlossenen Kreis*, dem *Regelkreis*, statt.

Als Beispiel sei ein Ofen genannt, dessen Temperatur konstant gehalten werden soll. Die Energiezufuhr, beispielsweise der Gasfluss, kann über ein Ventil (*Leistungssteller*) variabel gehalten werden. Unter der Voraussetzung, dass sich die äußeren Bedingungen nicht ändern, kann über die *Steuerung* der Stärke des Gasflusses eine bestimmte Temperatur im Ofen erzielt werden.

Ändern sich die äußeren Bedingungen um den Ofen (beispielsweise Schwankungen der Raumtemperatur oder der Außentemperatur, Öffnen eines Fensters oder einer Tür), existieren also *Störgrößen*, wird sich ein anderer als der erwartete Wert für die Ofentemperatur einstellen. Soll diese nun ihren konstanten Wert beibehalten, obwohl sich die äußeren Bedingungen mit einem nicht konstanten und unvorhersehbaren Verhalten ändern, ist eine *Regelung* erforderlich. Diese kann über ein Thermometer erfolgen, welches den Ist-Wert des Ofens (*Regelgröße*) erfasst. Abweichungen der Ofentemperatur können nun erkannt werden und durch den Bediener über die Energiezufuhr an den Soll-Wert (*Führungsgröße*) angeglichen werden. Die Energiezufuhr ist somit nicht mehr als starrer Wert vorgegeben, sondern ist mit der Temperatur des Ofens verknüpft. Hierbei tritt der Bediener in die Funktion eines *Reglers*. Man spricht von einem geschlossenen Regelkreis. Die Unterbrechung der Leitung des Temperaturfühlers resultiert in einem *offenen Regelkreis* und es liegt aufgrund der fehlenden Rückmeldung des Ist-Wertes nur noch eine *Steuerung* vor [184].

4.3.2.2 Qualität einer regeltechnischen Anlage

Die Regelgüte von Regelkreisen kann durch das Zitat von RECKNAGEL [176] beschrieben werden:

„Eine *Regeleinrichtung* hat die Aufgabe, die *Regelgröße* x möglichst genau auf einen vorgegebenen, zeitlich konstanten oder veränderlichen Wert zu halten und den Einfluss von *Störgrößen* auf ein Mindestmaß zu beschränken. Außerdem ist zu fordern, dass das Ausregeln [...] in kurzer Zeit erfolgt. Weiterhin sollen sich [...] *Überschwingungen* innerhalb bestimmter Grenzen befinden. Die Erfüllung dieser Forderungen ist nicht möglich, so dass hierbei Kompromisse geschlossen werden müssen.“

Die Probleme bei einer Optimierung der Regelgüte eines Regelkreises veranschaulicht die Abbildung 4.3.

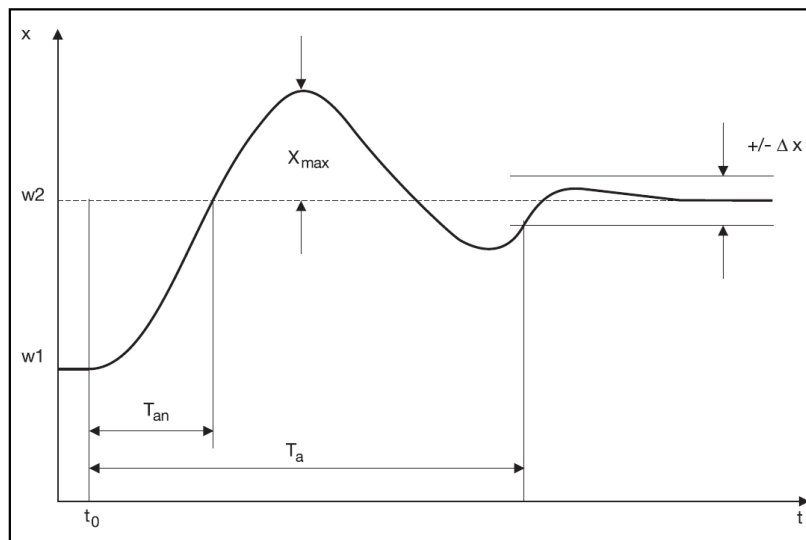


Abbildung 4.3: Regelgüte von Regelkreisen

Dargestellt ist die *Überschwingweite* X_{\max} , die *Anregelzeit* T_{an} , die *Ausregelzeit* T_a sowie der je nach Anforderungsbereich frei definierbare *Toleranzbereich* Δx . Wird die Dämpfung vergrößert um die Überschwingweite möglichst klein zu halten, wird zwangsweise die Anregelzeit vergrößert. Bei schnellen Anregelzeiten sind größere Überschwingungen der Regelgröße unvermeidbar.

In der Praxis müssen hierbei aufwendige und dadurch kostenintensive manuelle Parametrisierungen der Regeleinrichtungen, basierend auf lokalen Messdaten, durchgeführt werden. Dieser Sachverhalt zeigt das enorme Potenzial von computergestützter Parametrisierung und Optimierung für die praktische Anwendung.

Auswirkungen, die eine falsch parametrisierte Regelgröße für das Gesamtverhalten eines Regelkreises verursacht, werden in der Abbildung 4.4 veranschaulicht.

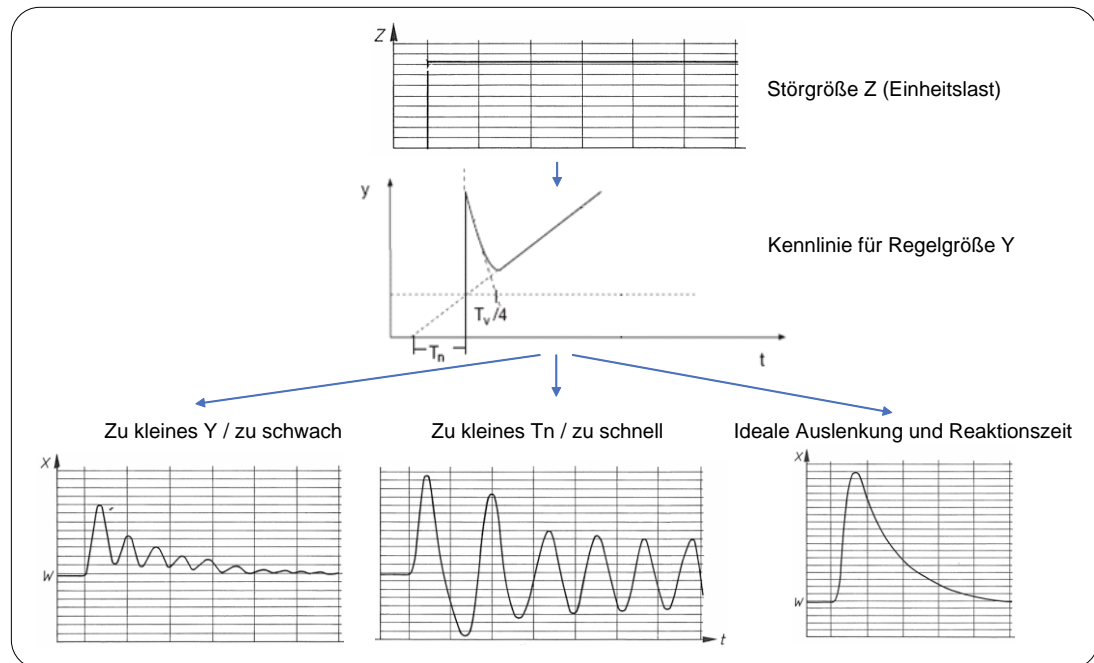


Abbildung 4.4: Einfluss der Regelparameter: Reaktion der Regelgröße Y auf eine sprunghafte Änderung der Störgröße Z

In der Abbildung 4.4 wird die mögliche Reaktion der Regelgröße Y auf eine sprunghafte Änderung der Störgröße Z beschrieben. Wird die Regelgröße zu klein gewählt, ist das Zeitfenster zum Erreichen der Führungsgröße (Soll-Wert) zu groß. Wird die Nachstellzeit T_n zu klein gewählt, können starke Fluktuationen im Regelkreis auftreten, welche eine Instabilität des Gesamtsystems hervorrufen können. Dieses Beispiel unterstreicht die Bedeutung der Aufgaben eines Regelungstechnikers bei der Optimierung von Regelwerken klimatechnischer Systeme. Nach SCHLEICHER [184] werden die wichtigsten Aufgaben für Regelungstechniker wie folgt dargestellt:

- Festlegung der Regelgröße
- Prüfung der Vorteile einer Regelung
- Festlegung des Messortes sowie Feststellung der Störgrößen
- Wahl des Stellgliedes und eines geeigneten Regelgerätes
- Positionierung der Regelgeräte
- Inbetriebnahme, Parametrisierung, Optimierung

Über den im Rahmen dieser Arbeit realisierten virtuellen Entwurfsraum in einem Verbund mit dem HVAC-Agenten Framework ist eine vollständig virtuelle Durchführung der hier aufgeführten Aufgaben einschließlich der geometrisch optimalen Anordnung der Anlagenkomponenten möglich.

4.3.2.3 Regelwerke

Das *Regeln* ist ein Vorgang, bei dem die fortlaufend erfasste *Regelgröße* mit einer *Führungsgröße* verglichen und im Sinne einer Angleichung an die *Führungsgröße* beeinflusst wird. Der sich daraus ergebende Wirkungsablauf findet im *Regelkreis* statt. Bei allen Regelkreisen ist das unterschiedliche Eingreifen des Reglers bei einer Abweichung der Regelgröße vom Sollwert von Bedeutung (schnell oder langsam, stark oder schwach). Dies hängt von den regelungstechnischen Eigenschaften der Regelstrecke ab, weshalb das statische und dynamische Verhalten der einzelnen Elemente (*Glieder*) des Regelkreises genau bekannt sein muss. Die Wahl eines passenden Regelgliedes ist abhängig vom Anwendungsfeld. In der Praxis kommt eine Vielzahl von Regelwerken zum Einsatz, welche abhängig von ihrer analogen oder digitalen Bauart unterschiedliche Charakteristiken aufweisen und als *unstetige* sowie *stetige Regler* klassifiziert werden können [56, 176, 184].

4.4 Software Agenten

Der in der deutschen Sprache als „*Agent*“ übersetzte Begriff eines Softwareansatzes zur Realisierung verteilter Systeme kann durchaus zu Irritationen führen, da eine Analogie zu IAN FLEMMINGS Spion *James Bond* in keiner Weise existiert. Vielmehr soll mit diesem Wort ein *Vermittler* oder *Vertreter* verstanden werden, welcher in Form einer Softwareumsetzung, durch implementiertes Fachwissen, vom Benutzer erhaltene Aufgaben selbstständig löst. Bislang existieren keine standardisierten Definitionen des Agentenkonzeptes. WOOLDRIDGE und JENNINGS [217] beschreiben einen Agenten folgendermaßen:

”Ein unabhängiges Programm, das in der Lage ist, seine Entscheidungen und sein Handeln, basierend auf der Wahrnehmung seiner Umwelt, bei der Verfolgung von Zielen selbstständig zu kontrollieren.“

Entscheidende Merkmale eines Agenten sind *Reaktivität*, *zielgerichtetes Verhalten*, *Autonomie* und *soziale Fähigkeit* [217]. Diese beschreiben die Fähigkeiten eines Agenten, auf Umwelteinflüsse reagieren (*Reaktivität*, *passive Aktivität*) sowie auch ohne äußere Einflüsse eine Funktion erfüllen zu können (*Proaktivität* oder *zielgerichtetes Verhalten*). Eine erweiternde Eigenschaft bildet hierbei die Autonomie eines Agenten (*Eigenkontrolle*), welche damit den grundsätzlichen Unterschied zu herkömmlichen Softwareprogrammen bildet. Um dieses Merkmal zu erfüllen, benötigt der Agent grundlegende Softwareansätze wie *Schlussfolgerung* und *Lernfähigkeit* aus seinem Ursprungsbereich, der (*Verteilten*) *Künstlichen Intelligenz* (VKI). Die *sozialen Fähigkeiten* eines Agenten beschreiben die Möglichkeit der Kommunikation mit anderen autonomen Systemen und mit dem Benutzer über die Bereitstellung notwendiger Schnittstellen.

Der Einsatz der Agenten-Technologie ermöglicht die Erstellung großer verteilter Systeme, bestehend aus selbständig agierenden Teilsystemen ohne zentrale Kontrollinstanz (siehe Abbildung 4.5).

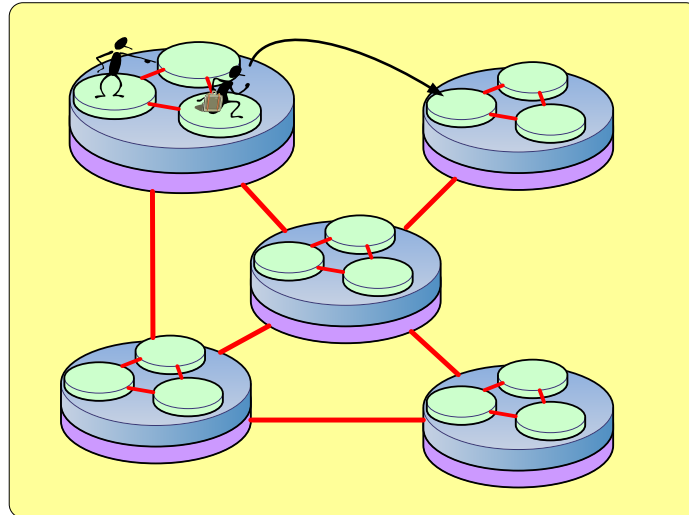


Abbildung 4.5: Die Agententechnologie ermöglicht die Realisierung verteilter Systeme ohne zentrale Kontrollinstanz

Die Agenten-Technologie gewinnt aufgrund der zunehmenden Vernetzung von Rechnern immer mehr an Bedeutung. Software-Agenten, welche auch bei inhomogenen Rechnerstrukturen und variierenden Bandbreiten innerhalb der Infrastruktur eine hohe Effizienz aufweisen, können einen lockeren Rechnerverbund zur Erzeugung eines verteilten Systems nutzen [178].

Simulationen über Software-Agenten sind in den Bereichen Biologie, Soziologie und Verkehrsforschung zum besseren Verständnis komplexer Zusammenhänge weit verbreitet. Ebenfalls wird die Agenten-Simulation auch verstärkt in der 3D-Computergrafik und Filmtechnik zur Visualisierung von Gruppenszenen eingesetzt. Auch auf dem Gebiet des Bauingenieurwesens werden Agenten zunehmend eingesetzt, hier verstärkt für die numerische Simulation [143, 169]. Der im Rahmen dieser Arbeit vorgestellte innovative Ansatz einer HVAC-Simulation innerhalb einer Computational Steering Umgebung beschreibt hingegen ein neues Anwendungsfeld für Software-Agenten.

4.5 Umsetzung

Unter den Gesichtspunkten der objektorientierten Programmierung kann ein Software-Agent als gekapseltes Objekt definiert werden. Statische Elemente eines objektorientierten Modells, wie Klassen, Attribute, Operationen und Assoziationen, werden hierbei durch ein dynamisches Verhalten der agentenorientierten Modellierung erweitert. Die *Agenten-Plattform* wird innerhalb einer Software über eine eigene Laufzeitumgebung realisiert. Hierbei existieren zahlreiche Werkzeuge zur Modellierung und Simulation von Agentensystemen. Bekannte Entwickler-Werkzeuge für Agenten-Frameworks sind unter anderem JADE [26], FIPA-OS [98], SWARM [43] und das ZEUS-TOOLKIT [46]. Für weiterführende Informationen zur Historie von Software-Agenten und einer Einführung in die „*agentenorientierte Software*“ wird auf [113] verwiesen.

Bei der Abbildung der HVAC-Agenten unterlag die Wahl der Entwicklungsumgebung aufgrund der engen Verknüpfung mit der CAD-Software AUTODESK *Architectural Desktop* (ADT) [16] starken Einschränkungen. Die C++ *Klassenbibliothek Qt* [209] erwies sich hierbei als gute Wahl zur Realisierung der geforderten Aufgaben. Qt bietet Interaktionsmöglichkeiten mit Simulationskern sowie CAD-Software und bietet über seine *BUS-orientierte*²⁵ Topologie eine ideale Voraussetzung für die dynamische Verwendung²⁶ von HVAC-Elementen.

Ein Hauptaspekt bei der Entwicklung des vorgestellten Software Prototyps für eine effiziente, vernetzt-kooperative Zusammenarbeit zwischen den einzelnen Fachplanern lag in der *Benutzerfreundlichkeit* der Anwendung. Das HVAC-Agenten Framework integriert sich ohne eine Unterbrechung des Arbeitsablaufes nahtlos in die vorhandenen Module des Prototyps. Der Wechsel zu einer zusätzlichen Software-Applikation neben dem ADT ist nicht erforderlich. Die Agenten wurden in Form von geometrischen Objekten in den AUTODESK Architectural Desktop implementiert und typenspezifisch in ADT-eigenen Bibliotheken zusammengefasst (siehe Abbildung 4.6 rechts). Der Repräsentant einer HVAC Komponente definiert sich hierbei über ein *AEC-Objekt* (siehe Abschnitt 2.4 sowie Kapitel 3) und gleicht grundsätzlich in Verhalten und Funktion den herkömmlichen geometrischen Konstruktionsobjekten des ADT. Da Agenten von AEC-Objekten abgeleitet werden, erben sie auch komplexe geometrische Eigenschaften wie die Objekt-Objekt-Beziehung über Ankerelemente, erweiterte Datenhaltung von Randbedingungen über Eigenschaftssätze und zeichnungsspezifische Darstellungsmöglichkeiten über *Display Representations* (siehe Abschnitt 3.2.2.2 und [15-17]).

²⁵ Ein BUS definiert im Bereich der Datenverarbeitung ein Leitungssystem zum Austausch von Daten zwischen mehr als zwei Teilnehmern. Die an einem allgemeinen Bus angeschlossenen Komponenten werden als Knoten oder Busteilnehmer bezeichnet. Hierbei existiert nur ein Daten-BUS, auf den die Knoten selbständig zum Schreiben oder Lesen von Daten zugreifen können. Innerhalb der Qt Klassenbibliothek wird dieses Verfahren als *Signal-Slot-Technik* bezeichnet. Ein Objekt im Sinne der objektorientierten Programmierung kann hierbei ein *Signal* senden, also Daten an eine globale Datenleitung übertragen. Über den *Slot*, einer Funktion zum spezifischen bzw. gefiltertem Abhören dieser Datenleitung, können angeschlossene Objekte entscheiden, ob sie die gesendeten Signale zur weiteren Verarbeitung empfangen.

²⁶ Die dynamische Verwendung bezieht sich auf das Hinzufügen und Entfernen von HVAC Komponenten sowie die Kopplung dieser Elemente während der Laufzeit der Simulation.

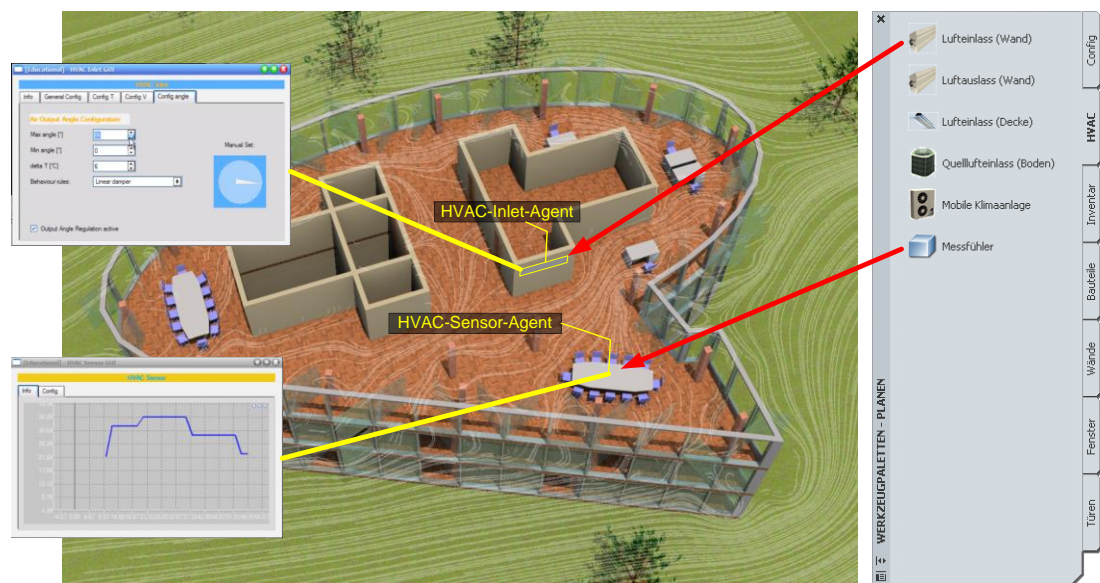


Abbildung 4.6: Implementierung der HVAC-Agenten in Form von geometrischen Objekten innerhalb des Autodesk Architectural Desktops. Auswahl der typenspezifischen Agenten über ADT-eigene Bibliotheken (rechts). Modifikation der Parameter für Regelwerke der Agenten über grafische Benutzeroberflächen (links).

Der Fachplaner hat die Möglichkeit, einen Agenten während des Simulationsvorganges über eine Bibliothekspalette in die Zeichnung einzufügen und innerhalb der Konstruktion anzuordnen, zu verschieben oder zu löschen. Das Verhalten der Agenten, wie beispielsweise Mess- und Regelintervalle, oder die Parameter der Regelwerke können über die typenspezifische grafische Benutzeroberfläche (GUI, siehe Abbildung 4.6 links) des jeweiligen Agenten verändert werden. Über ein Echtzeitdiagramm sind Analysen während des Simulationslaufs über das Verhalten einzelner Komponenten möglich.

Für die Auslegung einer Klimaanlageinstallation stehen dem Fachplaner verschiedene Bauelemente zur Verfügung. Die Abbildung 4.7 gibt eine Auswahl verschiedener Lufteinlässe beispielhaft wieder.



Abbildung 4.7: Darstellung unterschiedlicher Lufteinlässe einer Klimaanlageinstallation (Quelle: TROX-Technik [210])

4.5.1 Funktionsweise der Agenten

Die HVAC-Agenten können bei einer Klassifizierung in Agententypen den *Wrapper-Agenten* [11] zugeordnet werden. Die Abbildung 4.8 stellt den schematischen Aufbau der Agenten dar.

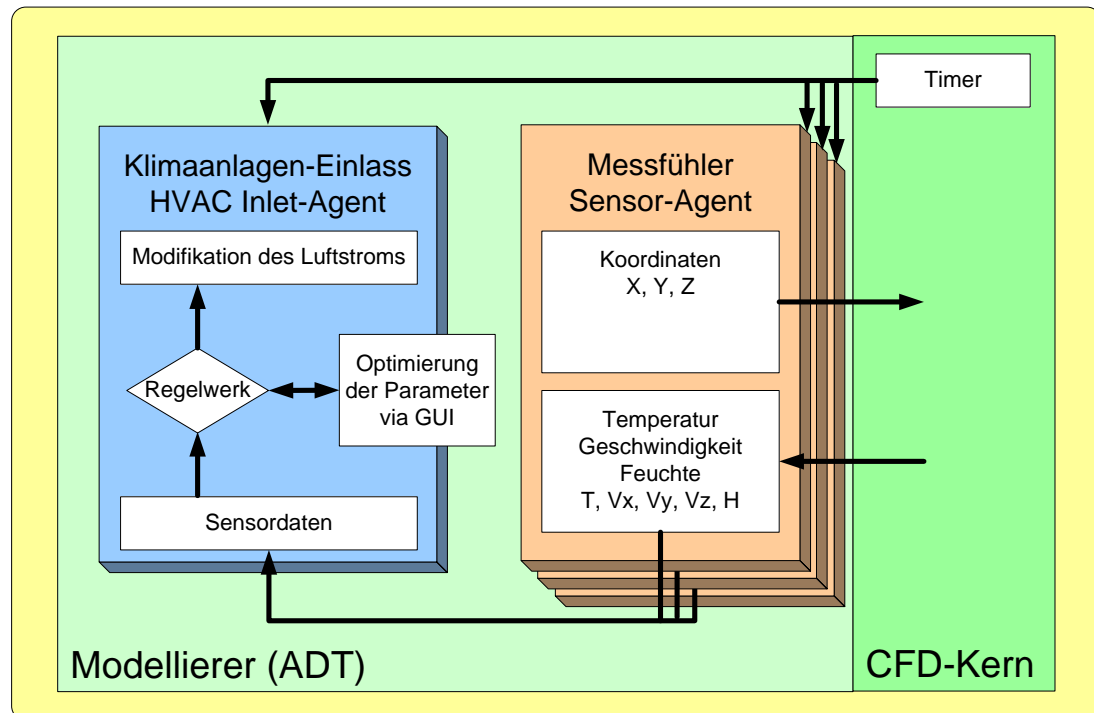


Abbildung 4.8: Schematischer Aufbau der Funktionsweise der HVAC-Agenten

Hierbei kann zwischen zwei Arten von Agenten unterschieden werden: Der *Sensor-Agent* dient der Abbildung von Messfühlern (für Temperatur, Luftfeuchte und Luftgeschwindigkeit), der *Inlet-Agent* emuliert verschiedene Arten von Lufteinlässen einer Klimaanlageinstallation. Darüberhinaus ist der Agent auch mit Verhaltensweisen zur Steuerung des Luftstroms über Regelwerke programmiert. Seine Struktur weist somit eine erhöhte Komplexität gegenüber dem Sensor-Agenten auf.

Der in der Abbildung 4.8 dargestellte Sensor-Agent besitzt Informationen über die Raumkoordinaten seines geometrischen Repräsentanten innerhalb des virtuellen Entwurfsraums. Diese Informationen werden bei der Erstellung oder Verschiebung des gekoppelten geometrischen Objektes aktualisiert. In Intervallen ermittelt der Sensor-Agent die an seiner Position herrschenden Klimabedingungen über eine Anfrage an den Rechenkern.

Der strukturelle Aufbau des Inlet-Agenten beinhaltet Informationen zu verknüpften Sensor-Agenten, ein Regelwerk und Funktionen zur aktiven Steuerung des Luftstroms. In regelmäßigen Abständen werden Messdaten von einem oder mehreren Sensor-Agenten durch den Inlet-Agenten abgefragt und über das implementierte Regelwerk verarbeitet. Abhängig von den Eingangsgrößen sowie den Parametern des Regelwerkes werden Lufttemperatur, Luftgeschwindigkeit, Luftfeuchte und Ausfallswinkel des Luftstroms neu berechnet und an das ADT übertragen. Parameter der Regelwerke zur Optimierung des

Raumklimas können über eine grafische Benutzeroberfläche des jeweiligen Inlet-Agenten vom Regeltechniker während der Laufzeit der Simulation modifiziert werden.

Alle Vorgänge zur Messung und Regelung des Luftstroms innerhalb der Computational Steering Umgebung erfolgen zeitabhängig. Somit ist ein Zeitgeber für diese Prozesse erforderlich, welcher nach einer Anzahl von Rechenschritten über den CFD-Kern einen Zeitimpuls an die Agenten des HVAC Frameworks übermittelt. Für einen effizienten Kommunikationsaustausch zwischen Agenten Framework und CFD-Kern werden die Anfragen der Software-Agenten an den Rechenkern über eine Kontrollinstanz, den *Observer*, gesammelt und gebündelt übertragen. Analog wird der Zeitimpuls des Rechenkerns direkt an den Observer gesendet und an die Agenten verteilt.

Die Sensor- und Inlet-Agenten speichern die Werte von Messungen, Regelparametern und Luftstromgrößen über den gesamten Simulationslauf. Für eine nachfolgende Analyse durch den Regeltechniker können diese Daten über eine Exportschnittstelle in eine fachplanerspezifische Anwendungssoftware übertragen werden.

4.5.2 Eingliederung in das Computational Steering Konzept

Das HVAC-Agenten Framework bietet eine ideale Voraussetzung zur Integration in die Funktionsweise des Computational Steering Ansatzes. Die Abbildung 4.9 stellt die Kopplung zwischen dem Simulationsprozess des Prototyps und den Software-Agenten dar.

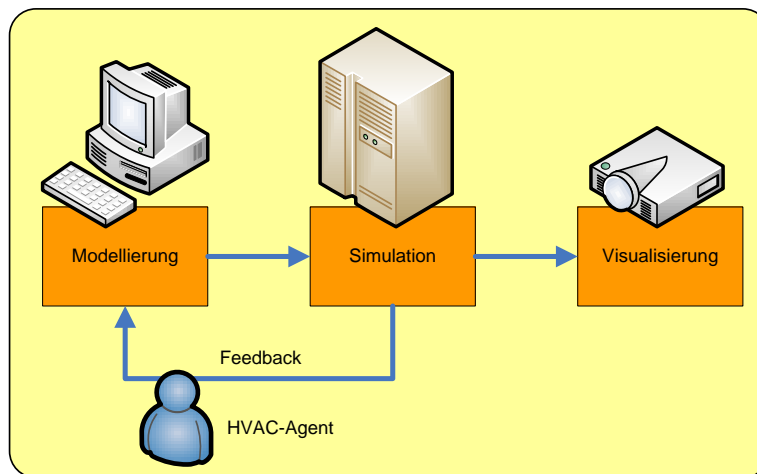


Abbildung 4.9: Integration des HVAC-Agenten Framework in das Konzept des Computational Steering

Durch einen über die Agenten initiierten Rückkopplungseffekt kann der interaktive Simulationsprozess ohne Eingriffe des Benutzers aktiviert werden. Über eine diametrale Betrachtungsweise kann die Behauptung aufgestellt werden, dass die Vorteile der Agenten-Technologie erst über die Einbindung in eine Computational Steering Umgebung in ihrer vollen Funktion genutzt werden können.

4.5.3 Anbindung des Agenten Frameworks an das ADT

Das HVAC-Agenten Framework wird als eigene C++-Applikation (HVACCOMClient) realisiert. Aufgrund der spezifischen Anforderungen an die Software, bezogen auf eine effiziente Kommunikation mit dem Rechenkern, und das parallele Ausführen mehrerer Prozesse (*Threads*, [209]) ist eine vollständige Integration in das ADT nicht möglich. Als Schnittstelle zwischen Agenten Frameworks und ADT dient die von MICROSOFT entwickelte COM/COM+ *Technologie* [146], welche über die MICROSOFT *Foundation Classes* (MFC) [147] angesprochen wird.

Die Kommunikation erfolgt hierbei über eine *COM-Bridge* (HVACCOMBridge, siehe Abbildung 4.10). Sie besitzt für einen beidseitigen Informationsaustausch zwischen Agenten Framework und virtuellem Entwurfsraum eine *bidirektionale* Funktionalität.

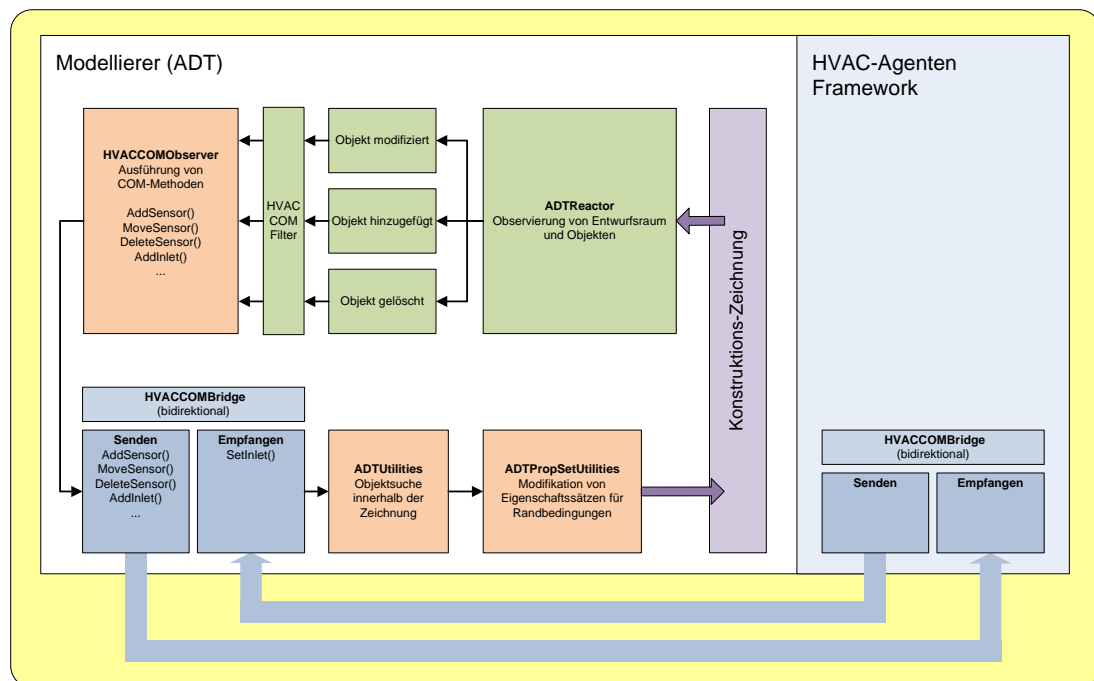


Abbildung 4.10: Schematische Darstellung der Schnittstellen für den interaktiven Datenaustausch zwischen dem ADT und dem HVAC-Agenten Framework

Über diese Schnittstelle ist es innerhalb des virtuellen Konstruktionsraumes möglich, die von einem *Reactor* (siehe Abschnitt 3.3) des ADT detektierten agentenspezifischen Änderungen wie das Hinzufügen, Löschen oder Bewegen einer Klimaanlage-komponente, an das HVAC-Agenten Framework weiterzugeben. Ebenfalls können über Regelwerke berechnete Simulationsgrößen wie Lufttemperatur, Luftgeschwindigkeit oder Einfallswinkel vom Agenten Framework an den ADT zur automatischen Speicherung in den Eigenschaftssätzen des jeweiligen AEC-Objektes (siehe Abschnitt 3.3) übertragen werden.

4.6 Klimaregelung über HVAC-Agenten

Die in Abschnitt 4.3.2.3 beschriebenen Regelwerke definieren das Regelungsverhalten einer Regelgröße. Diese bezieht sich auf eine Führungsgröße unter Einfluss von einer oder mehreren Störgrößen. Eine optimale Auslegung der Parameter für ein Regelwerk einer klimatechnischen Installation bestimmt neben der geometrischen Positionierung die Qualität einer Klimaanlageinstallation.

Der HVAC Inlet-Agent beinhaltet gängige Regelglieder aus der Klasse der *stetigen Regler* [56]. Hierüber lassen sich Regelwerke über die Mehrzahl von Reglern abbilden, welche auch in der Praxis Anwendung finden. Es wurden *proportionale Regler* (*P-Regler*), *integrierende Regler* (*I-Regler*) und *differenzierende Regler* (*D-Regler*) implementiert, welche in ihrer Kombination *P*-, *I*-, *PD*-, *PI*- und *PID-Regler* [56] ergeben (siehe Abbildung 4.11).

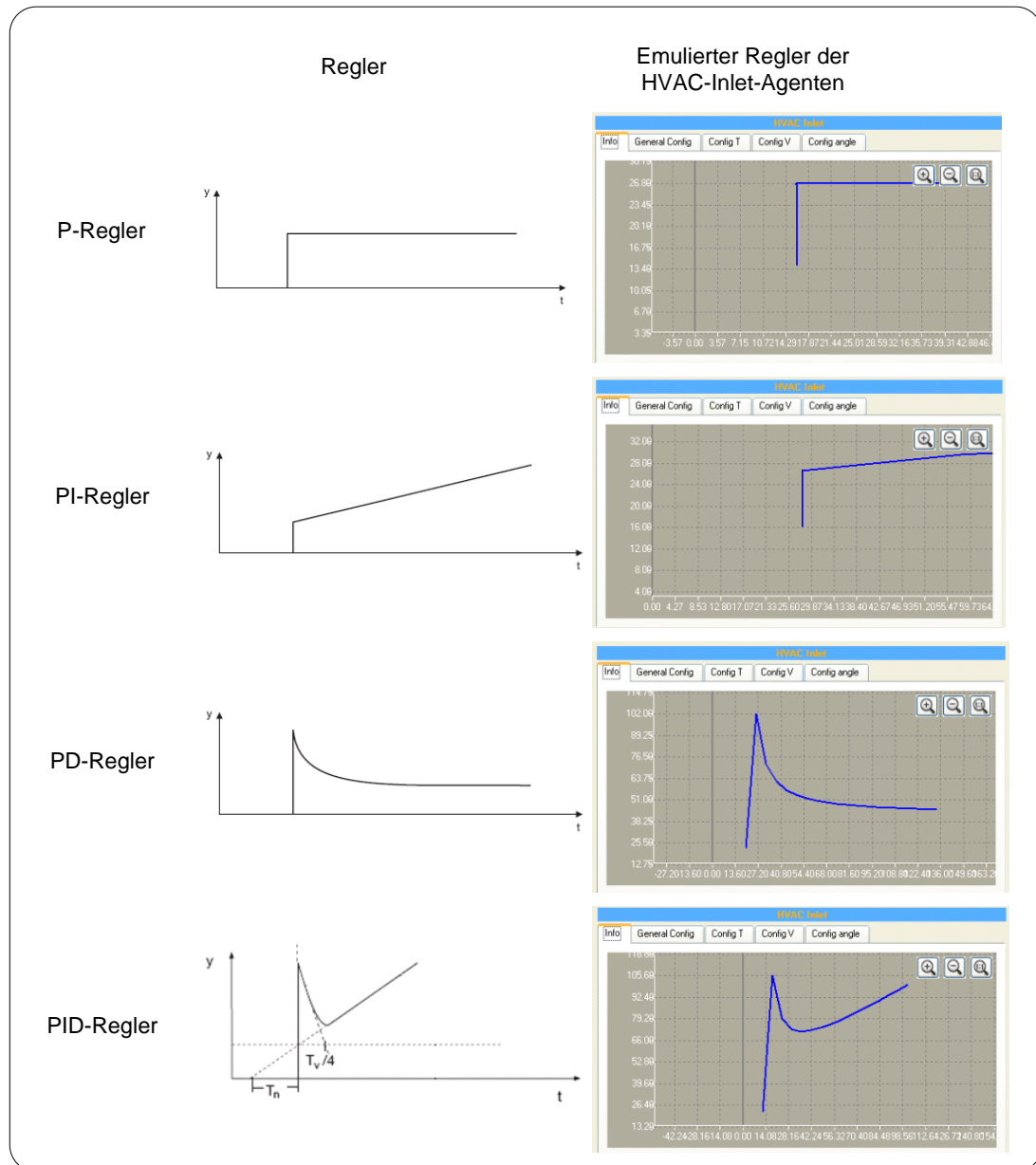


Abbildung 4.11: Die Kennlinien eines HVAC-Systems werden über Regelglieder bestimmt. Der linke Teil der Abbildung stellt die Referenz für das Verhalten des Regler aus der Literatur dar. Der rechte Teil der Abbildung zeigt eine grafische Ausgabe der emulierten Regelglieder der HVAC-Agenten.

Entsprechende Elemente können innerhalb des Simulationslaufs über eine grafische Oberfläche vom Hausleittechniker ausgewählt und über Parameter optimiert werden (siehe Abbildung 4.12).

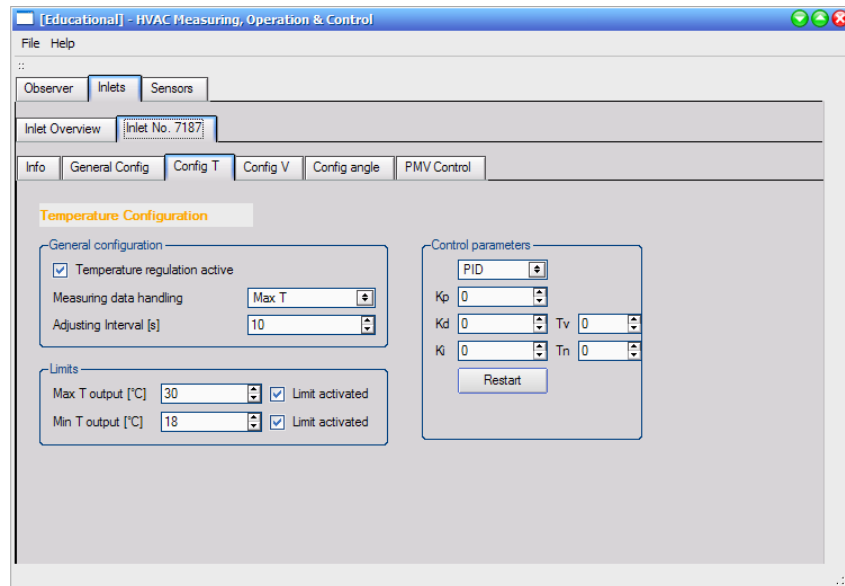


Abbildung 4.12: Grafische Benutzeroberfläche (GUI) eines Inlet-Agenten für die Modifikation von Parameter zu Regelwerken, Reaktions- und Antwortzeiten. Die Einstellungen können für jede Simulationsgröße spezifisch durchgeführt werden.

Der Fachplaner erhält die Möglichkeit, die Parameter der Regelstärke, der Reaktionszeit und der Vor- und Nachlaufzeiten sowie die Parameter der Regelglieder zu Temperatur, Geschwindigkeit und Ausfallswinkel individuell zu bestimmen und diese mit einem *Behaglichkeitsfeld* (siehe Abschnitt 4.2 und 4.6.3) zu verknüpfen. Für die Analyse werden Mess- und Regeldaten von Sensoren und Auslässen in Echtzeit über eine Plot-Funktion ausgegeben (siehe Abbildung 4.11 rechts).

Durch selektives Aktivieren der P-, I- und D-Glieder lassen sich über ihre Kombination Regler für unterschiedliche Regelaufgaben erstellen: Die Temperaturregelung erfolgt im Allgemeinen über PID-Regler, für die Regulierung von Durchflüssen (Luft- und Wasserströme) dienen I-Regler. Um eine Klappensteuerung zu realisieren, wie sie bei den Klimaanlageauslässen eingesetzt wird, kommen P-Regler für eine direkte Regelumsetzung zum Einsatz [184].

4.6.1 Übertragungsverhalten

Der Simulationskern überträgt zwei Arten von Zeitimpulsen, *Simulationszeit* und *simulierte Zeit*. Die *Simulationszeit* bezeichnet die Zeit, die der CFD-Kern für eine Berechnung benötigt, also die *real ablaufende Zeit*. Die *simulierte Zeit* beschreibt den Zeitraum, welcher in der *virtuellen Welt* vergeht - also simuliert wird - und ist beispielsweise abhängig von der Auflösung des Rechengebietes und der Rechenleistung des PC-Clusters. Als Beispiel werden über einen Optimierungslauf Luftströmungen innerhalb eines Raumes simuliert. Die Strömungen, welche zehn Minuten nach Starten des Simulationslaufs dargestellt werden, beschreiben nicht die Strömungen, wie sie in der virtuellen Welt nach zehn Minuten in Erscheinung treten. Der Benutzer erhält eine Darstellung der Strömungsverläufe nach beispielsweise vier Stunden. Der Rechenkern kann in der Regel die Entwicklung von Luftströmungen schneller berechnen, die Darstellung läuft sozusagen im „Zeitraffer“ ab und ist nicht synchron mit der Zeit, welche in der realen Welt vergeht.

Für das Übertragungsverhalten der Zeitimpulse an das Agenten Framework, mit der sich die Agenten zur Steuerung der Regelkreise synchronisieren, ist die *simulierte Zeit* von Bedeutung. Abhängig von den Simulationskriterien leitet der CFD-Kern die virtuell vergangene Zeit über die Anzahl der Berechnungsschritte ab und überträgt diese, wie in der Abbildung 4.13 dargestellt, an das HVAC-Agenten Framework.

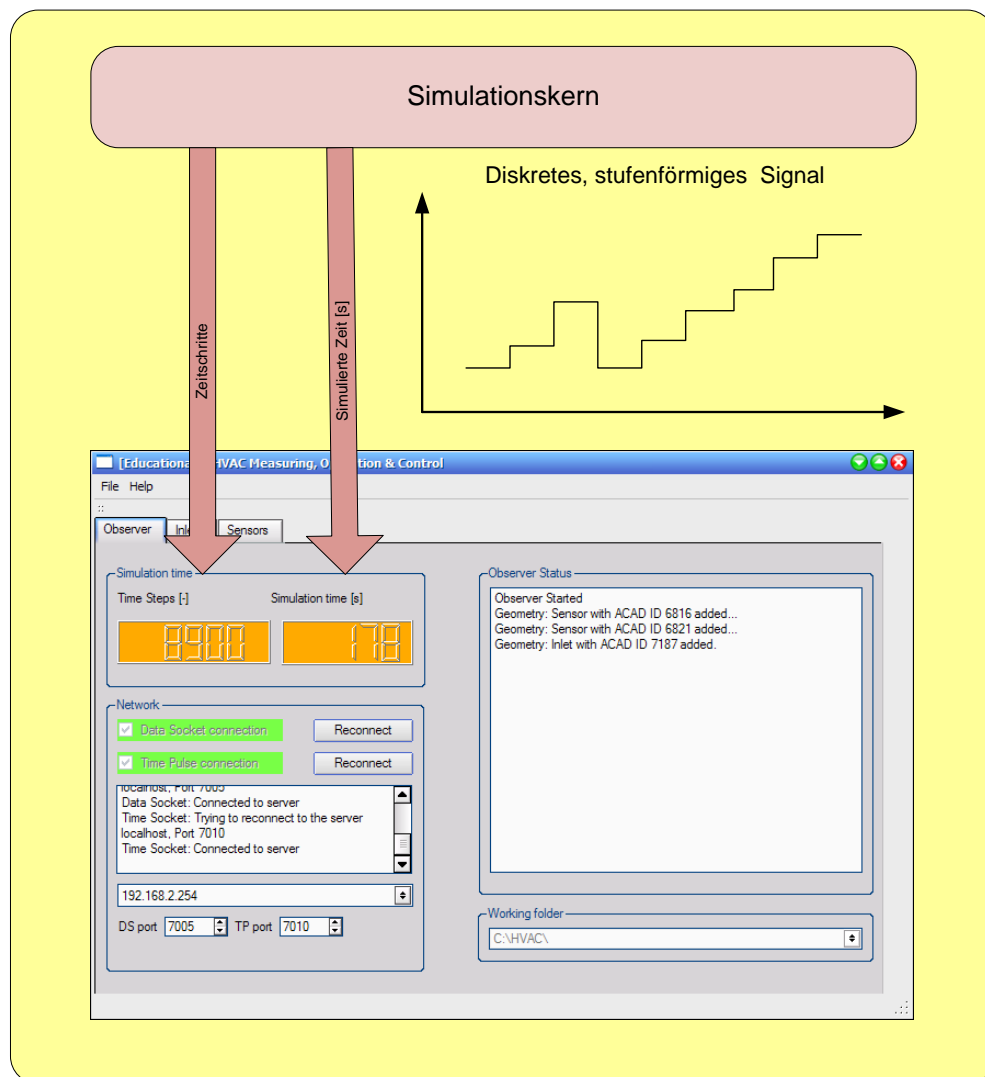


Abbildung 4.13: Übertragung des Zeitimpulses vom Simulationskern an die HVAC-Agenten zur Steuerung der Regelwerke.

Die dabei übertragenen digitalen Signale gehören der Gruppe der *diskreten Signale* an. Der zeitliche Verlauf eines diskreten Signales weist Stufen auf. Nach DIN 19 225 [56] kann die zu verarbeitende Signalform als *wertkontinuierliches* und *wertdiskretes*, *zeitkontinuierliches* und *zeitdiskretes* Signal klassifiziert werden. Demnach ergeben sich unter Berücksichtigung von Wert und Zeit vier verschiedene Arten der Signalformen. Das an die Agenten übertragene Signal weist ein wertkontinuierliches und zeitdiskretes Verhalten auf.

Bei den Regelwerken der HVAC-Agenten bestimmen die *Regelalgorithmen* das Übertragungsverhalten. Hierbei ist unter anderem ein *Stellungs-Algorithmus*, bei dem das Ausgangssignal jeweils zum Abtastzeitpunkt gebildet wird und ein *Geschwindigkeits-Algorithmus*, bei dem die Ausgangssignaländerung vom vorangegangenen zum aktuellen Abtastzeitpunkt, also der Stellgeschwindigkeit, gebildet wird, implementiert. Für weiterführende Informationen zum Aufbau dieser Algorithmen, im Speziellen für PID-Regler, wird auf die DIN 19 225 [56] und auf [176, 184] verwiesen.

4.6.2 Interaktive Optimierung eines Reglers

Die optimale Auslegung eines Regelwerkes ist entscheidend für die Stabilität, Reaktionsgeschwindigkeit und den Energieverbrauch einer Klimaanlageinstallation. Somit muss der korrekten Auslegung der Parameter eines Regelwerks ein hoher Stellenwert angerechnet werden. Die in der Literatur genannten Formulierungen zur Optimierung eines Reglers sind unscharf und beruhen auf Erfahrungswerten. Die Durchführung des Optimierungsvorganges erfolgt hierbei nach der Installation einer Anlage über langwierige Untersuchungen. Da somit kein idealisiertes Optimierungsschema existiert, sind diese Vorgänge in der Regel mit einem hohen Zeit- und Kostenaufwand verbunden.

Beispielhaft soll im Folgenden das praktische Vorgehen zur Optimierung eines PID-Reglers ohne Kenntnis der Regelstrecke erläutert werden. Der Prozess erfolgt hierbei in Teilschritten (siehe Abbildung 4.14). Initial wird die P-Struktur des Reglers aktiviert. Die anfänglich hoch gewählte Stellgröße wird schrittweise reduziert, bis sich ein stabiler Endwert nach zwei bis drei Schwingungen mit einer bleibenden Regelabweichung einstellt (Abbildung 4.14, a). Nachfolgend wird die D-Struktur des Reglers zur Justierung der Dämpfung aktiviert (Abbildung 4.14, b). In einem letzten Schritt kann nun der I-Anteil hinzu geschaltet werden um über die PID-Struktur eine Erreichung des Sollwertes zu gewährleisten (Abbildung 4.14, c) [184].

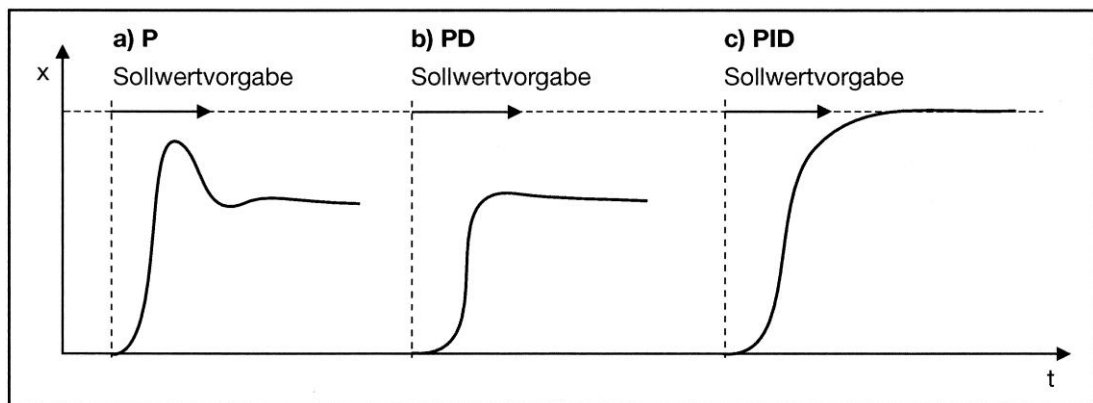


Abbildung 4.14: Notwendige Teilschritte zur optimalen Auslegung eines PID-Reglers [184]

Dieses Beispiel zeigt, dass die Parametrisierung und Optimierung eines Reglers über den Regeltechniker einen hohen Aufwand verursacht und über mehrere Teilschritte durchzuführen ist. Über die HVAC-Agenten kann der hier beschriebene Vorgang vollständig virtuell durchgeführt werden, da P-, I- und D-Regelglieder unabhängig von einander aktiviert und parametrisiert werden können.

4.6.3 Klimaregelung über ein Behaglichkeitsfeld

Die Behaglichkeit im Raum wird über den *PMV-Index* (*Predicted Mean Vote*; [106, 177]) definiert. Dieser sagt einen Durchschnittswert für die Klimabeurteilung durch eine große Personengruppe über eine siebenstufige Klimabeurteilungsskala voraus (siehe Abschnitt 4.2). Primären Einfluss hat hierbei der temperierte Luftstrom der Klimaanlageauslässe, welcher über die implementierten Regelkreise dynamisch angepasst wird.

In der Praxis hat sich die Regelung der Klimaanlagesteuerung über *Behaglichkeitsfelder* etabliert. Hierbei wird nicht der *optimale Behaglichkeitswert* (PMV-Index 0 = neutral) als Zielgröße der Regelung eingesetzt, sondern vielmehr über Temperatur und Luftfeuchtigkeit ein *Bereichsfeld* gewählt, welches für die definierte Personengruppe als akzeptabel gilt (siehe Abbildung 4.15).

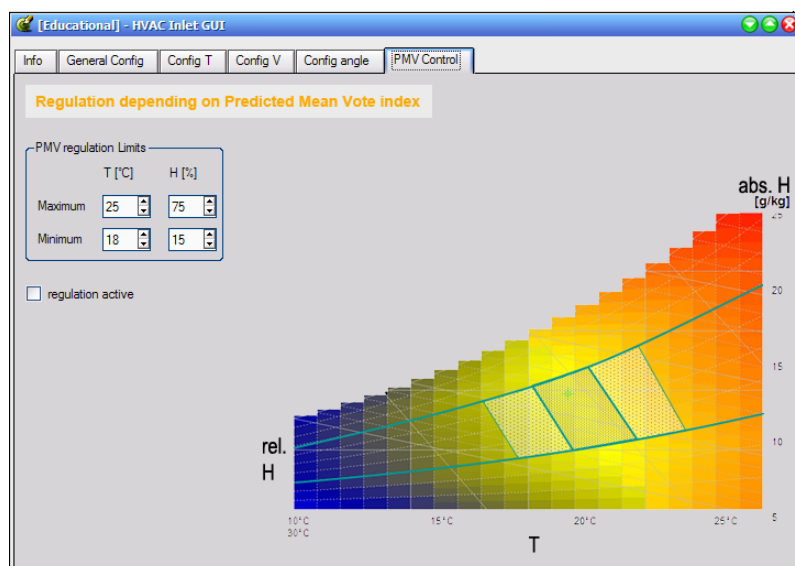


Abbildung 4.15: Grafische Benutzeroberfläche (GUI) des Inlet-Agenten zur Festlegung der Grenzen eines Behaglichkeitsfeldes

Befinden sich die Temperatur- und Feuchtigkeitswerte innerhalb des definierten Feldes, wird die Anlage über die Steuertechnik deaktiviert. Erst bei einem Überschreiten der Grenzen des Behaglichkeitsfeldes wird die Anlage zur Nachsteuerung reaktiviert. Durch dieses Verfahren sind bei modernen Klimainstallationen enorme Energieeinsparungen möglich.

Die hier vorgestellten HVAC-Agenten des Prototyps unterstützen die Regelung über Behaglichkeitsfelder (siehe Abbildung 4.15). Die Regelwerke der Inlet-Agenten können mit einem frei definierbaren Behaglichkeitsfeld gekoppelt werden und ermöglichen somit die Optimierung der Energiekosten über eine Simulation.

Interaktive Optimierung von Regelparametern. In der Literatur existieren verschiedene empirische Verfahren zur Auslegung der Parameter bestimmter Arten von Regelwerken. Ein Verfahren basiert auf dem Ansatz, die Stellgröße periodisch mit einer anschließenden Analyse der Regelgrößenänderung zu ändern. Dieser Prozess wird mit unterschiedlichen Schwingungsfrequenzen der Stellgrößen durchgeführt. Damit kann aus Amplitude und Phasenverschiebung der Regelgrößenschwankung der Frequenzgang der Strecke ermittelt werden. Über dieses *schematische* Verfahren lassen sich die Regelparameter explizit ableiten [184]. Da dieser Prozess mit einem sehr hohen mathematischen Aufwand verknüpft ist, kann er in der Praxis nicht eingesetzt werden.

Erfolgt hierbei die Durchführung über einen numerischen Lösungsansatz unter Nutzung des HVAC-Agenten Frameworks, kann dieses Verfahren als einfache Stapelverarbeitung zum Ablauf mehrerer verknüpfter Simulationsläufe mit variierenden Parametern betrachtet werden. Der Implementierungsaufwand für diesen Mechanismus ist gering.

Hilfsmittel für die Unterstützung von HVAC/Haustechnik-Installationen, wie beispielsweise die automatische Erstellung von Lüftungskanälen werden vom ADT nicht unterstützt. Ebenfalls fehlt die Anbindung an Produktdatenbanken der HVAC-Hersteller, bei denen handelsübliche HVAC Komponenten mit entsprechenden Spezifikationen²⁷ für den Konstruktionsprozess eingesetzt werden können. Die auf *AutoCAD* basierende Software *Autodesk Building Systems* [21], welche speziell für den Bereich Elektrotechnik und Hausleittechnik entwickelt wurde, bietet neben diesen genannten Funktionalitäten auch eine umfangreiche Datenbank von HVAC Elementen. Gegenüber dem ADT besitzt diese Software ein erweitertes Produktmodell mit direkter Unterstützung von HVAC Elementen über Eigenschaftssätze. Die *Autodesk Building Systems* Software wird nur für die Region Nordamerika entwickelt und die dazugehörige Bauteilbibliothek beinhaltet nur HVAC-Bauteile für den nordamerikanischen Markt. Eine für den deutschen Markt nutzbare, weniger umfangreiche Alternative bietet die Software *RoCAD* [152], welches als Plug-In über das ADT eingesetzt wird. Diese Software stellt bei einer Einbindung in den virtuellen Entwurfsraum eine mächtige Erweiterung für den praktischen Einsatz dar. Doch würde diese aufgrund der Inkompatibilität hinsichtlich des IFC Produktmodells und der OMF Schnittstelle zu starken Einschränkungen für das Gesamtkonzept des Prototyps führen.

Nähere Kopplung des HVAC-Agenten Frameworks an das IFC-Produktmodell. Das in Kapitel 2 beschriebene IFC Produktmodell beinhaltet eine eigene HVAC Domain. In einem zukünftigen Schritt kann eine Erweiterung hinsichtlich der Datenhaltung für die HVAC-Agenten erfolgen. Hierbei sollen die Property Sets für Metainformationen von HVAC Komponenten welche keinen Standards entsprechen, auf IFC-konforme Property Sets umgestellt werden.

Als Beispiel sind die von der IFC standardisierten Eigenschaftssätze *IfcAirTerminalBoxType* und *IfcAirTerminalType* zu nennen. Diese beinhalten Leistungsdaten einer Anlage, wie die *AirFlowrateRange* oder die *TemperatureRange* für Minimal- und Maximalwerte der Lufttemperatur und Luftgeschwindigkeit im Operationszustand. Diese Eigenschaftssätze können über einen *Parser* direkt in das Agenten Framework überführt werden. Weitere Attribute der IFC HVAC Domain (wie

²⁷ beispielsweise Hersteller, Ausstattung, Leistung und Kosten

MountingType und CoreType) geben Aufschluss über die Bauart der Anlage und würden die eindeutige Identifizierung und Zuordnung eines HVAC-Agententyps erlauben [97].

Über das IFC Produktmodell besteht die Möglichkeit, HVAC-Agent sowie deren Metainformationen über IFC Objekte abzubilden. Ein HVAC Sensor wird hierbei als Element vom Typ `IfcSensorType` gespeichert. Die IAI unterscheidet hierbei zwischen verschiedenen Messsensoren, wie `TemperatureSensor`, `HumiditySensor` und `PressureSensor`. Die über das HVAC-Agenten Framework genutzte Kombination mehrerer Messwerte pro Sensor (Feuchte, Temperatur, Druck und Geschwindigkeit des Luftstroms) kann hierbei über eine Aufteilung in spezifische IFC-Sensoren erfolgen [97].

Überführung der HVAC Ergebnisdaten aus der Simulation in ein IFC Produktmodell. Bei der Entwicklung des HVAC-Agenten Framework lag die eindeutige Zielsetzung auf der Implementierung eines Computational Steering Systems zur Durchführung eines interaktiven, kooperativen Optimierungsprozesses. Somit ergab sich konzeptionell nicht die Notwendigkeit, Ergebnisdaten der Simulation als Datensatz in ein Produktmodell zu überführen. Dennoch wäre eine datenkonforme Umsetzung hinsichtlich des IFC Produktmodells gegeben:

Die IFC ermöglicht die Datenhaltung von Leistungsdaten einer Klimaanlage-komponente über das Property Set `Pset_AirTerminalBoxPHistory` der Entity `IfcPerformanceHistory`. Diese Funktionalität ist für einen Vergleich von realen Messwerten bei Baufertigstellung gegenüber Messwerten aus den Simulationen der Planungsphase sinnvoll. Zusätzlich sieht die IAI die Datenhaltung zonenspezifischer Messungen vor. Die Speicherung von Messungen über die klimatische Behaglichkeit geometrisch definierter Arbeitsplätze eines Raumes ist hierüber möglich. Dies ermöglicht die Konvertierung der Daten in das Property Set `Pset_SpaceThermalPHistory` (ebenfalls von der Entity `IfcPerformanceHistory`) für Lufttemperatur, Luftfeuchte und Luftgeschwindigkeit. Die von den Sensor-Agenten generierten Messreihen können hierbei als `IfcTimeSeries` in das Produktmodell überführt werden [97].

Anforderungen an den Rechenkern für die HVAC-Simulation. Bei der Optimierung der klimatischen Behaglichkeit innerhalb eines Raumes werden Komponenten mit sehr geringen Lufteinlassgeschwindigkeiten für die Auslegung der Klimaanlagekonstruktion gewählt. Gewöhnlich ist eine Rasteranordnung vieler kleinen Komponenten förderlicher für das Behaglichkeitsempfinden als der Verbau weniger großer Komponenten. Die aktuelle Entwicklung der *Quelllufteinlässe* (*Swirl diffusers*) [176, 210], welche Luft in einer radialen Bewegung mit extrem geringen Geschwindigkeiten einblasen, zeigt hier einen positiven Einfluss auf das Raumklima. Erst durch die Interaktion der Luft mit den im Raum befindlichen Personen findet durch eine Konvektionswirkung über die Oberflächentemperatur der Körper ein Lufttransport statt. Die geometrische Abbildung dieser Klimaanlagekomponenten für die Simulation weist einen Durchmesser von nur wenigen Zentimetern auf. Für realistische Ergebnisse muss das komplexe Verhalten des radial einblasenden Luftstroms vollständig abgebildet werden. Durch die enorme Größendifferenz zwischen Swirl und Gesamtgeometrie sind die Anforderungen an den Rechenkern hoch [131]. Ein adaptives Rechengitter mit lokaler Diskretisierung (beispielsweise in einem definierten Radius um die Auslässe) ist notwendig.

Kapitel 5

Eingesetzte Technologien des Software Prototyps

5.1 Virtual Reality

Ein Großteil der menschlichen Wahrnehmung erfolgt über visuelle Signale. Komplexe Zusammenhänge lassen sich über eine grafische Darstellung schneller und einfacher vermitteln als über eine Beschreibung von Sachverhalten. Der Erfolg der visuellen Darstellung ist hierbei eng verbunden mit den Mechanismen des menschlichen Lernens.

Über die *Virtuelle Realität* (oder engl. *Virtual Reality*) wird der Versuch unternommen, dem Betrachter über eine computergenerierte, virtuelle Umgebung die Darstellung einer Szene mit ihren physikalischen Eigenschaften als Wirklichkeit zu vermitteln. Der Benutzer versetzt sich hierbei in eine dreidimensionale Welt, in der er mit virtuellen Gegenständen interagieren kann. Er hat beispielsweise die Möglichkeit, Türen in einem Haus zu öffnen, Einrichtungsgegenstände zu verschieben oder ein Fahrzeug zu steuern. Die Virtuelle Realität nutzt hierbei Techniken der dreidimensionalen Modellierung sowie der Video- und Bildverarbeitung [40].

5.1.1 Stereoprojektion

Die Techniken der Virtuellen Realität beruhen auf der in Abbildung 5.1 dargestellten stereoskopischen Bilderzeugung [128, 136]. Sie bemächtigt sich zur Entstehung eines *immersiven* Eindrucks eines einfachen Tricks, um die visuelle Wahrnehmung des menschlichen Gehirns zu täuschen.

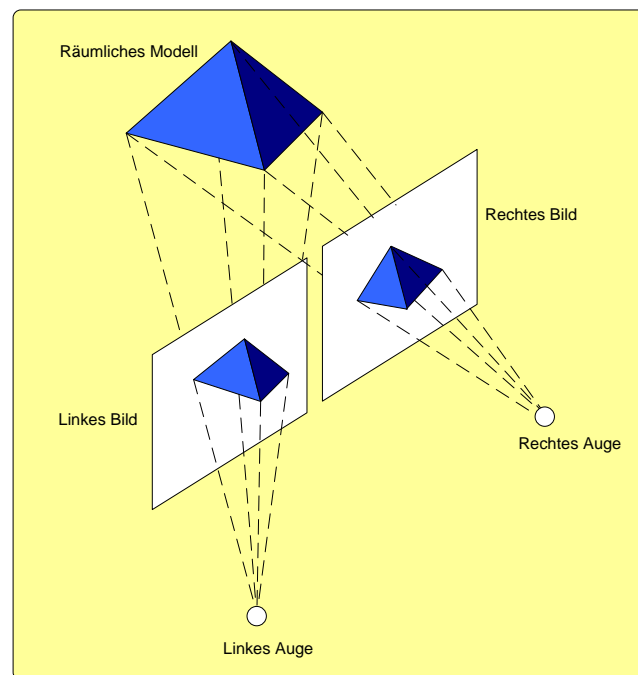


Abbildung 5.1: Durch eine optische Täuschung kann dem Betrachter der immersive Eindruck eines räumlichen, dreidimensionalen Abbildes einer Szene oder eines Modells vermittelt werden.

Die Möglichkeit der dreidimensionalen Wahrnehmung eines Menschen basiert auf dem Prinzip des *binokularen Sehens*. Dabei werden die Bildinformationen eines Objektes beiden Augen getrennt mit einer minimal variierenden Perspektive zugeführt. Diese beiden unterschiedlichen Bildinformationen eines Objektes (*Parallaxe*) nutzt das Gehirn zur Konstruktion eines dreidimensionalen Bildes. Durch die Differenz der beiden Perspektiven kann ebenfalls die Entfernung des Objektes vom Betrachter interpretiert werden (*Tiefenwirkung*). Bei geringen Blickwinkeländerungen liegt ein hoher Abstand zum Objekt vor, sind die perspektivischen Änderungen groß, ist das Objekt sehr nah.

Über die Aufteilung der Ansicht eines räumlichen Modells in zwei getrennte Perspektiven, welche als zweidimensionale Bilder (*stereoskopische Halbbilder*) jeweils getrennt dem linken und rechten Auge des Betrachters zugeführt werden, kann das Phänomen der *immersiven Wahrnehmung* hervorgerufen werden. Die generierten Halbbilder ergeben sich hierbei aus den Schnittpunkten der Sehstrahlen. Die Strahlen reichen vom räumlichen Modell bis zur Eintrittsöffnung der Augen und schneiden eine virtuelle Ebene im Raum (siehe Abbildung 5.1).

5.1.2 VR-Technologie

Bei einer VR-Umgebung unterscheidet man zwischen *Voll-* und *Teilimmersiven Systemen*. Diese können abhängig von der eingesetzten Hardware zur Darstellung der virtuellen Szene und der Anzahl der möglichen Benutzer der Anlage (Einzelplatz- oder Kooperative VR-Umgebung) in weitere Unterkategorien unterteilt werden. Zu den teilimmersiven Systemen zählt auch die *Augmented Reality*, bei dem eine virtuelle Szene in die reale Szene eingeblendet wird und dadurch beide Welten miteinander verschmelzen. Für die

Realisierung einer VR-Umgebung müssen vier grundlegende Elemente berücksichtigt werden [139]:

- Die Darstellung einer computergenerierten, dreidimensionalen Szene mit einem ausreichend hohen Grad an Realismus, um einen immersiven Effekt zu erzeugen.
- Die Interaktivität des Systems, welches für eine effiziente Benutzerinteraktion in Echtzeit reagieren muss.
- Die virtuelle Umgebung muss dem Benutzer das Gefühl vermitteln können, ganz oder teilweise mit der immersiven Welt verbunden zu sein.
- Das System muss über drei Dimensionen mit dem Benutzer zur blickwinkelabhängigen Anpassung der Szenenperspektive gekoppelt sein. Dieses erfolgt über Sensoren zur permanenten Positionserfassung des Benutzers.

Zur Darstellung solcher Szenen können unterschiedliche Hardwarelösungen herangezogen werden. Häufige Anwendung findet hierbei das *Head-Mounted-Display* (HMD, [41]), ein oder mehrere nebeneinander angeordnete Großbildleinwände mit planer oder gewölbter Oberfläche (*Powerwalls*, [73]) und die *CAVE*²⁸ [132].

Die Visualisierung der Szene über die beschriebene Hardware erfolgt nach dem Prinzip der in Abschnitt 5.1.1 beschriebenen *Stereoprojektion*. Hierbei erfolgt die Aufteilung der Bilder für das rechte und linke Auge nach dem in der Abbildung 5.2 dargestellten Verfahren.

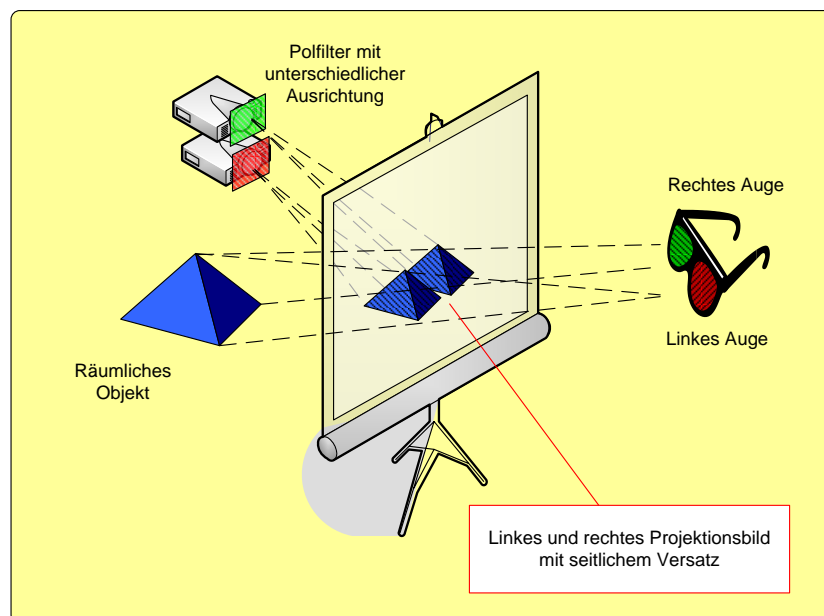


Abbildung 5.2: Funktionsweise einer Stereo-Projektionsanlage in einer Virtual Reality Umgebung

²⁸ CAVE = engl. „Cave Automatic Virtual Environment“ oder dt. „Höhle mit automatisierter, virtueller Umwelt“. Eine CAVE stellt einen würfelförmigen Raum dar, in dessen Mittelpunkt der Benutzer positioniert ist. Alle sechs Flächen des Würfels, somit Wände, Decke und Boden dienen als Projektionsflächen zur Darstellung der virtuellen Szene. Die CAVE stellt aktuell die bestmögliche Lösung für die Darstellung einer immersiven Welt dar.

Für die Darstellung einer virtuellen Szene werden zwei Projektoren benötigt, welche beide Modellperspektiven überlagernd auf einer Leinwand abbilden. Neusten Projektions-systemen gelingt die Darstellung dreidimensionaler Bilder mit nur einer Optik [55].

Die Bildtrennung kann über *aktive* oder *passive Verfahren* erfolgen. Für die in der Abbildung 5.2 dargestellte passive Trennung werden auf den Linsen der beiden Projektoren, welche das jeweilige Teilbild für das linke und rechte Auge abbilden, *Polfilter* mit unterschiedlicher Ausrichtung (in der Regel um 90° versetzt) installiert. Der Betrachter trägt eine Brille, welche mit zwei Polfiltern unterschiedlicher Ausrichtung ausgestattet ist. Das Bild eines Projektors wird somit nur über das Brillenglas mit gleich ausgerichtetem Polfilter an das jeweilige Auge übertragen.

Für ein *aktives Verfahren* zur Bildtrennung wird die *Shutterbrille* eingesetzt, welche mit zwei Flüssigkristallanzeigen ausgestattet ist. Die Brille kann über eine elektronische Steuerung in hoher Frequenz (in der Regel 60 Hz) das rechte und linke Sichtfenster der Brille in Wechselschritten verdunkeln. Durch die Synchronisation zwischen Brille und Projektoren wird innerhalb einer Frequenz das Bild eines Projektors nur einem Auge zugeführt. Das andere Auge ist innerhalb dieser Frequenz über das Brillenglas vollständig verdunkelt.

Spezielle *Eingabegeräte* dienen der Interaktion des Benutzers mit der virtuellen Umgebung. Zum Einsatz kommen hierbei sensorische Systeme wie *optische und magnetische Tracker*, *Datenhandschuhe* oder *Datenanzüge*, *Biosensoren*, *Wands* (sog. Zauberstäbe) und die *Spacemouse*, welche dreidimensionale sowie axiale Bewegungen erfassen kann [12, 89].

5.1.3 Einsatzgebiete

Die Idee einer Virtuellen Realität hat ihren Ursprung in den vierziger Jahren in der Entwicklung eines Flugsimulators für den Trainingseinsatz von Piloten. Die 1965 von SUTHERLAND veröffentlichte Publikation „*The Ultimate Display*“ [198] beschreibt einen Helm zur Darstellung einer immersiven dreidimensionalen Szene und gilt als Grundstein der Entstehung der Virtuellen Realität.

In der heutigen Zeit stellt die Entwicklung von Systemen zur Darstellung immersiver Umgebungen ein wachsendes Anwendungsfeld hinsichtlich der Optimierung der *Mensch-Computer-Interaktion*²⁹ dar. Die industriellen und wissenschaftlichen Einsatzgebiete liegen in den Bereichen der Produktentwicklung, Datenanalyse, Einsatzplanung und Ausbildung. Bekannte Beispiele sind Flugsimulatoren zur Pilotenausbildung und die Entwicklung von Prototypen in der Automobilindustrie. VR-Anlagen sind häufig mit *Computational Steering Systemen* verknüpft [134].

Im Bereich des Bauwesens findet die Virtuelle Realität vergleichsweise geringe Anwendung. Neben der Visualisierung von Gebäuden [170] werden sie im Bereich der Strömungsmechanik [138, 139], bei der energetischen und ökologischen Leistungsanalyse von Gebäuden [135], der Strukturanalyse [173] und der Projektplanung [133] eingesetzt.

²⁹ HCI = engl. **H**uman-**C**omputer-**I**nteraction

5.2 Computational Steering

Der Ablauf einer typischen numerischen Simulation erfolgt in mehreren Teilschritten (siehe Abbildung 5.3, oberer Teil). Dabei definiert der Benutzer über das *Pre-Processing* die Anfangs- und Randbedingungen des Simulationslaufs und übergibt diese an den Simulationskern. Erst nach einem vollständigen Durchlaufen der Simulation werden die Ergebnisdaten ausgegeben und können im abschließenden *Post-Processing* über ein Analyse- oder Visualisierungstool ausgewertet werden. Ist das erhaltene Ergebnis aufgrund von ungünstig gewählten Parametern für den Benutzer nicht zufriedenstellend, ist eine Wiederholung aller Teilschritte notwendig. Das Erreichen der Zielfunktion erfolgt somit über einen iterativen Prozess zwischen Pre- und Post-Processing mit langen Antwortzeiten.

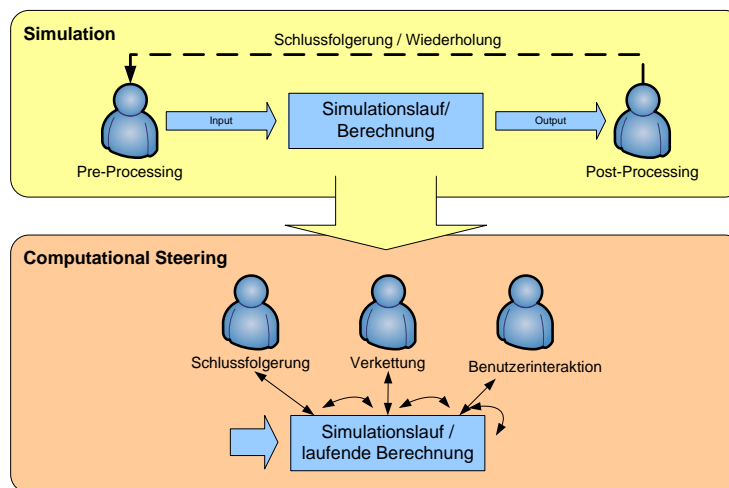


Abbildung 5.3: Vergleich zwischen dem typischen Verlauf einer Simulationsberechnung und dem Verlauf einer Simulation über ein Computational Steering System

Über ein *Computational Steering System* (siehe Abbildung 5.3, unterer Teil) werden nunmehr interaktive Mechanismen zur Verfügung gestellt, welche Simulation, Pre- und Post-Processing miteinander vereinen. Dem Benutzer stehen die Möglichkeiten der Visualisierung, Datenanalyse und Nachbearbeitung *zur Laufzeit* der Simulation zur Verfügung. Über diese Kopplung können die Antwortzeiten eines Simulationslaufs und somit der Iterationsprozess zum Erreichen der Problemlösung stark beschleunigt werden. Die direkte Interaktion mit dem System erhöht das Verständnis für Zusammenhänge zwischen den Modifikationen der Randbedingungen und der Geometrie und dem daraus resultierenden Verhalten der numerischen Berechnung. Der englische Begriff *steering* bezieht sich sinngemäß auf das direkte *Steuern von Parametern* zur Laufzeit der Simulation bei einer zeitnahen Visualisierung der Ergebnisdaten.

Die Realisierung eines solchen Systems erfolgt häufig über die Kopplung mit einer Virtual Reality Anlage zur dreidimensionalen Visualisierung. Zur Erzielung akzeptabler Antwortzeiten des Simulationskerns werden bei dreidimensionalen Problemen hohe Anforderungen an die Rechenleistung des Systems gestellt. PC-Cluster finden hier häufigen Einsatz [71]. Die Entwicklung von Computational Steering Systemen ist Gegenstand aktueller Forschungsprojekte [65, 117, 187]. Erste halb-kommerzielle Prototypen sind in der Entwicklung [214], obwohl noch kein industrieller Standard abzusehen ist.

5.3 Vernetzt-kooperative Planung

Die Integration der Software VNC [175] erlaubt die Realisierung einer Multi-User-Umgebung für einen kooperativen und verteilten Zugriff mehrerer Fachplaner auf ein gemeinsames Display. Die Fensterapplikationen des Modellierers und der Visualisierungssoftware können dadurch an eine beliebige Anzahl von Client-Rechner übertragen werden (siehe Abbildung 5.4).

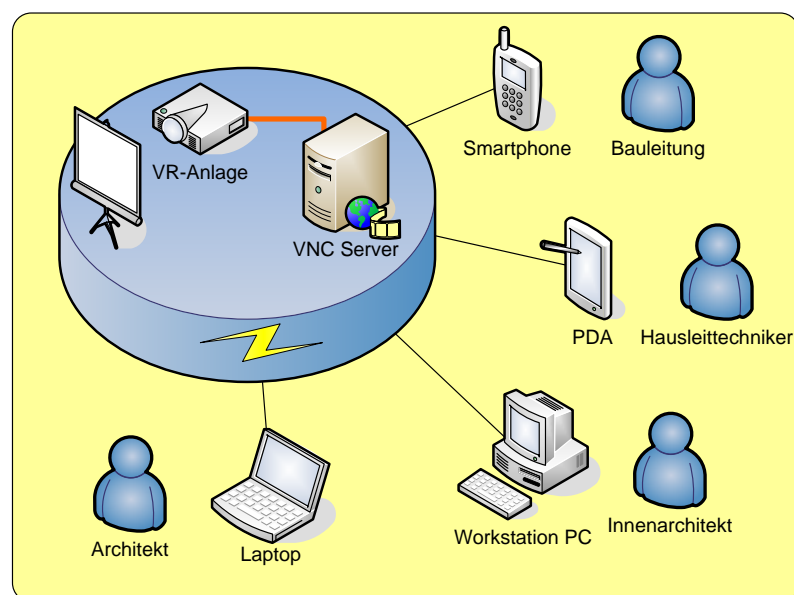


Abbildung 5.4: Realisierung einer Multi-User-Umgebung über die Software VNC

Grafische Änderungen werden in Echtzeit aktualisiert. Hierbei wird die Kontrolle über Maus und Tastatur für die Applikationen des Prototyps jeweils einem Benutzer zur Steuerung des Prozesses übertragen.

Das Übertragungsprinzip unterscheidet sich gegenüber objektorientierten Ansätzen wie sie bei der Middleware *CORBA*³⁰ [165] (siehe Projekt *CoCoS* [33]) oder bei der agentenbasierten Kommunikation [216] zum Einsatz kommt. Die Grundlage der Datenübertragung dieser Technologien basiert auf dem Senden einzelner Entwurfsobjekte von einem zentralen Server an die angebundenen Client-Rechner. Die Objekte werden über die Software des Client-Rechners interpretiert, in die lokale Kopie des Gesamtmodells integriert und visualisiert. Die im Rahmen dieser Arbeit angewendete Software überträgt den Bildschirminhalt der Fensterapplikationen des Servers als pixelorientierte Information an die Client-Rechner. Das Prinzip ähnelt der Verteilung eines Datenstroms (*Streaming*) für eine Videoübertragung. Die Nachteile dieser Software zeigen sich durch die reduzierte Interaktion mit dem System, welche gegenüber den objektorientierten Ansätzen auf einen gleichzeitigen Benutzerzugriff beschränkt ist. Hierbei erlauben objektorientierte Ansätze ein simultanes Modifizieren unabhängiger Objekte durch selektives Sperren (*Object-blocking*). Vorteile durch eine globale Sperrung des Arbeitsbereiches über die VNC-Software zeigen sich hingegen durch das Vermeiden einer redundanten Datenhaltung und einem damit verbundenen Versionskonflikt. Einen zusätzlichen Vorteil bietet die Unterstützung verschiedener Softwareapplikationen [140] zur Übertragung des virtuellen Entwurfsraums an mobile Endgeräte, für beispielsweise die Nutzung auf der Baustelle.

Die Software VNC ist plattformunabhängig und kann ohne zusätzliche Software als *Java-Applet* [196] über eine Browser-Applikation gestartet werden. Eine verzögerungsfreie Bildübertragung ist nur bei einer Netzanbindung mit hoher Bandbreite gewährleistet. Die DSL- oder die mobile UMTS-Technik sind hierfür ausreichend.

Die Effizienz des kooperativen Planungsprozesses in einer Multi-User-Umgebung wird durch die Einbindung von Audio- und Videoübertragung erhöht. Die Übertragung erfolgt über die freie Software *Skype* [190], welche zusätzlich den Gebrauch mobiler Endgeräte unterstützt.

³⁰ **C**ommon **O**bject **R**equest **B**roker **A**rchitecture (CORBA) ist ein Projekt der OBJECT MANAGEMENT GROUP (OMG) welches plattformübergreifende Protokolle und Dienste definiert. CORBA vereinfacht das Erstellen verteilter Anwendungen in heterogenen Umgebungen. Die OMG ist ein Konsortium von über 700 Firmen welche das gesamte Spektrum der Computerindustrie repräsentieren.

Kapitel 6

Modelltransfer

6.1 Einleitung

Das primäre Ziel des Software Prototyps liegt in der Optimierung der Luftströme innerhalb von Gebäuden. Dies erfolgt über die Durchführung einer numerischen Simulation durch Fachplaner ohne Kenntnisse der Strömungsmechanik und Informatik. Zur Erfüllung dieses Anspruches wurde der Prozess des Pre-Processing, welcher zur Vorbereitung eines Modells für die numerische Simulation notwendig ist, als automatisierter Hintergrundprozess realisiert.

Das erforderliche Berechnungsmodell für die numerische Simulation ist prinzipiell inkompatibel zu dem als Datenbasis vorliegenden IFC Produktmodell. Somit erfolgt die in der Abbildung 6.1 dargestellte Modellaufbereitung über eine automatisierte Abfolge komplexer Teilschritte.

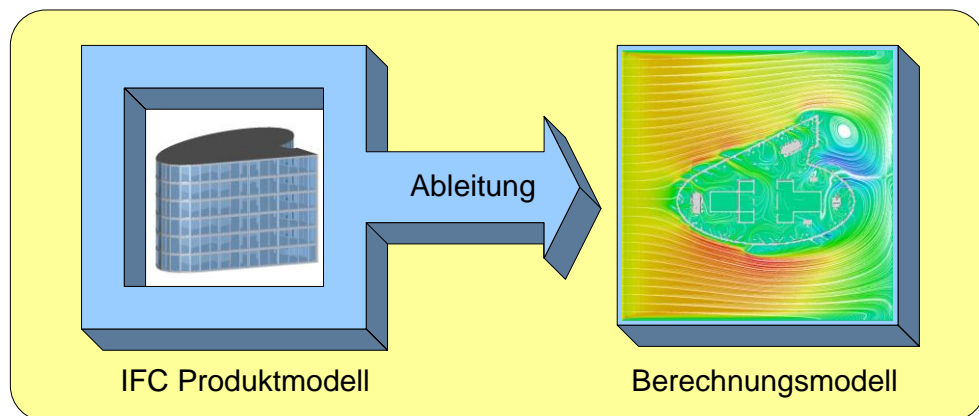


Abbildung 6.1: Ziel des automatisierten Modelltransfers ist die Ableitung eines Berechnungsmodells für die Strömungssimulation aus einem Produktmodell.

Die Durchführung dieser Konvertierungsschritte soll nach einer Einführung in die geometrische Modellierung in diesem Kapitel erläutert werden.

6.2 Dreidimensionale geometrische Modellierung

Bei der Erstellung und Darstellung von Plandaten im Bauwesen hat der Einfluss von rechnergestützter geometrischer Modellierung in den letzten 20 Jahren zunehmend an Bedeutung gewonnen. Die seitdem rapide fortschreitenden Entwicklungen von CAD-Systemen und Computer Hardware bieten nunmehr effiziente Möglichkeiten, neben den in der Praxis immer noch weit verbreiteten zweidimensionalen Repräsentationen auch dreidimensionale Geometriemodelle zu verarbeiten. Dieser dreidimensionale und volumenorientierte Ansatz eines geometrischen Modells ist die Grundlage des Rechenmodells für die numerische Simulation.

Es muss zwischen den Begriffen *Geometrie* und *Topologie* unterschieden werden. Die *Geometrie* definiert bei der Modellbildung die Gestalt von Körpern im Raum mit ihrer exakten Position und Form (Lage eines Punktes, gerade oder gekrümmte Verbindung zweier Punkte, Form von Flächen). Dabei können beispielsweise Oberflächen von Körpern als Polyeder beschrieben werden, welche aus Punkten, Linien und Flächen bestehen [36, 39]. Die nachbarschaftliche Beziehung zwischen diesen Elementen wird über die *Topologie* definiert. Diese Form der geometrischen Beschreibung wird somit für eine qualitative Lagebeziehung zwischen geometrischen Objekten eingesetzt und ist bei der Betrachtung von *Oberflächenmodellen* von entscheidender Bedeutung [42]. Im Folgenden werden die wichtigsten geometrischen Modelle erläutert, welche individuell oder verknüpft ihren Einsatz in modernen CAD-Systemen finden [74].

Wireframe Model (Drahtmodell). Das Drahtmodell ist eines der einfachsten Verfahren zur Darstellung dreidimensionaler Elemente. Die Repräsentation erfolgt ausschließlich über (gekrümmte) Kanten. Diese Technik wird von CAD-Systemen zur Erstellung geometrischer Schnitte genutzt und benötigt nur wenig Datenspeicher sowie geringe Rechenleistung. Das Verfahren zeigt aber starke Nachteile bei der Volumenmodellierung. Grund hierfür sind unter anderem fehlende Flächeninformationen zum Eliminieren verdeckter Körperkanten, welche eine mehrdeutige Interpretation eines Körpers oder gar die Konstruktion widersprüchlicher Geometrie erlauben [39].

Constructive Solid Geometry-Modell (CSG). Geometrische Grundprimitive bilden die Basis für das CSG-Volumenmodell. Hierbei resultieren komplexe Körper als Ergebnis aus sukzessiven Mengenoperationen (Durchschnitt, Vereinigung und Differenz) sowie geometrischen Transformationen (Skalierung, Translation, Rotation) von Grundkörpern. Diese Grundprimitive werden beispielsweise über Quader, Zylinder, Kegel und Kugeln repräsentiert. Die Datenhaltung wird über einen Binärbaum beschrieben, bei dem die geometrischen Primitive die Blätter, die booleschen Operatoren die inneren Knoten und das resultierende Bauteil die Wurzel darstellen [39].

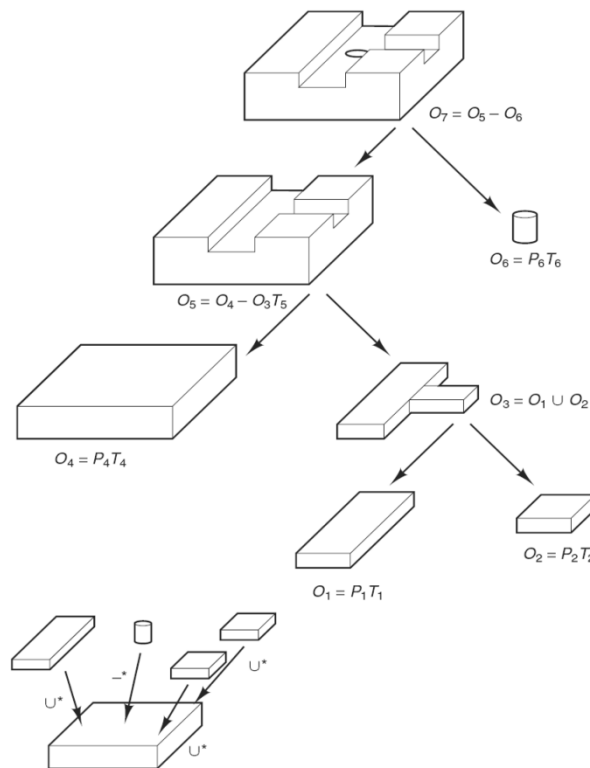


Abbildung 6.2: CSG Darstellung eines Körpers durch Mengenoperationen und Transformationen von Grundkörpern (Primitiven).

Das CSG-Modell wird allgemein nur im Verbund mit weiteren geometrischen Modellen für CAD-Systeme eingesetzt. Grund hierfür ist die fehlende exakte Beschreibung eines Bauteils über die Datenstruktur. Das Objekt muss aus einer impliziten Modellbeschreibung hergeleitet werden. Neben dem daraus resultierenden Rechenaufwand sind Methoden zum Vergleichen von Objekten nicht anwendbar, da eine eindeutige Datenstruktur zur Erstellung eines Körpers nicht existiert. Unterschiedliche Kombinationen von booleschen Operationen können mit unterschiedlichen Grundkörpern bei identischem Ergebnis durchgeführt werden. Die Ableitung von Draht-, Flächen- oder anderen Volumenmodellen aus der CSG-Datenstruktur ist nicht möglich.

Boundary Representation (B-rep) Oberflächenmodell. Das Prinzip der Visualisierung eines Oberflächenmodells basiert auf der Beschreibung eines Körpers über die Körperoberfläche, welche in eine endliche Menge von Teilflächen unterteilt ist. Diese Teilflächen werden über deren sie begrenzenden Kanten und Punkte beschrieben (siehe Abbildung 6.3).

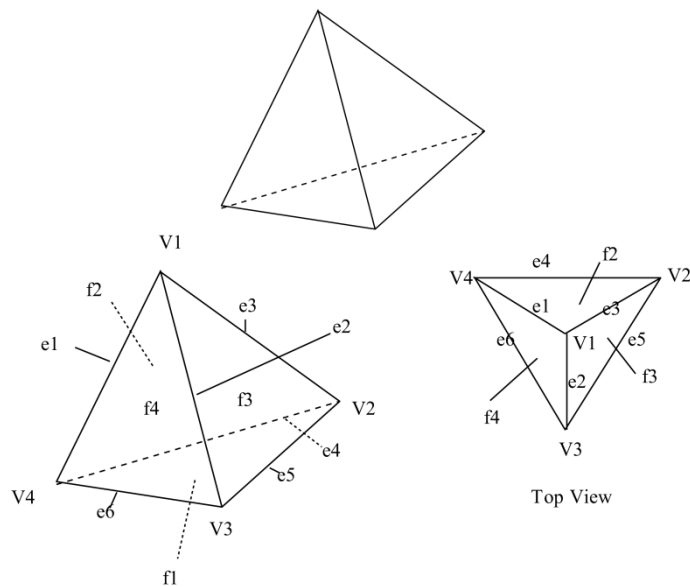


Abbildung 6.3: Das Oberflächenmodell beschreibt die Geometrie eines Körpers über Knoten, Kanten und Flächenstücke.

Neben den geometrischen Informationen enthält das Oberflächenmodell auch topologische Daten. Die topologische Struktur des geometrischen Modells wird über den *ref-Graph*³¹ G beschrieben, indem eine Menge von Eckpunkten oder Knoten V (*vertices*), Kanten E (*edges*) und Flächenstücken F (*faces*) über eine *Adjazenzrelation*³² R in Beziehung gesetzt werden (siehe Abbildung 6.4).

³¹ *ref* steht für **vertices** (Knoten), **edges** (Kanten) und **faces** (Flächenstücke).

³² Adjazenz beschreibt die Beziehung zwischen Knoten oder Kanten in einem ungerichteten Graph. Zwei Knoten sind adjazent oder benachbart, wenn sie durch eine Kante verbunden sind. Analog können zwei Kanten als adjazent bezeichnet werden, wenn sie sich an einem Knoten berühren.

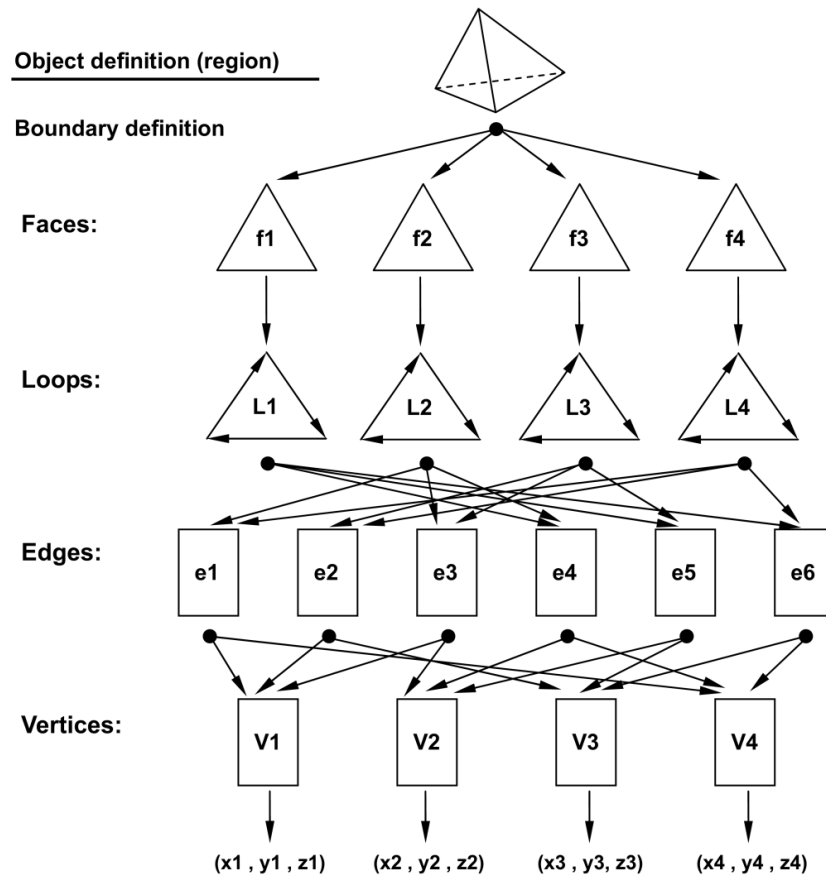


Abbildung 6.4: Die topologische Struktur des Oberflächenmodells beschreibt die Beziehungen von Knoten (vertices), Kanten (edges) und Flächenstücken (faces) über einen vef-Graph.

Somit erfolgt $G = (V, E, F, R)$, wobei $V := \mathbb{Z}^3$, $E := \{e_1, \dots, e_{nE}\}$ und $F := \{f_1, \dots, f_{nF}\}$ darstellt. Die Knoten von G sind die Elemente aus V , E oder F . Für einen detaillierten Einblick in die topologische Struktur von Oberflächenmodellen sowie die Bildungen von Relationen zwischen Elementen wird auf [39] verwiesen.

Das Oberflächenmodell bietet grundlegende Vorteile gegenüber dem CSG-Modell: Durch das explizite Vorhalten der Objektgeometrie kann ein direkter Zugriff auf die Datenelemente erfolgen. Die Oberflächeninformation eines über das B-rep Modell beschriebenen Körpers ist ebenfalls einzigartig. Somit ist eine eindeutige Identifizierung über Vergleichsmethoden möglich. Durch die Orientierung der Segmente über ihren Normalenvektor kann zwischen Außenseite und Innenseite des beschriebenen Körpers unterschieden werden. Durch Verwendung von *Freiformkurven* und *Freiformflächen* [39] bei der Beschreibung von Kurven und Flächen bei Oberflächenmodellen kann eine hohe mathematische Genauigkeit erzielt werden. Die Informationsdichte für Topologie und Geometrie kann abhängig vom Anwendungsfall variieren. Als Beispiel sei hier die Autoindustrie genannt, welche für die Geometrie einer Karosserieform eine Flächenbeschreibung mit hohem Detaillierungsgrad benötigt. Im Bereich des Bauwesens ist hingegen die geometrische Beschreibung einer großen Anzahl von Bauteilelementen mit einfacher Geometrie der topologischen Informationsdichte untergeordnet.

Der hier erläuterte Ansatz zeigt Nachteile hinsichtlich der Beschreibung von „ungünstigen“ Körpern [39] sowie der Komplexität der Datenstruktur in Bezug auf Speicherbedarf und Verarbeitungszeiten. Zur Optimierung der Datenstruktur des ursprünglichen Oberflächenmodells wurden Erweiterungen wie beispielsweise das *Half-Edge-Modell*, das *Winged-Edge-Modell* oder das *Radial-Edge-Modell* entwickelt [205].

Das B-rep Modell wird im Bereich der CAD-Anwendungen am häufigsten eingesetzt und findet auch innerhalb des AUTODESK *Architectural Desktop* Anwendung. Grundlage bildet hier der *AModeler* (siehe [19] und Abschnitt 6.5), welcher die Generierung, Verwaltung, Transformation und Visualisierung von facettierten B-rep Modellen unterstützt. Andere auf dem Markt vertretene Geometriekerne, wie beispielsweise *ACIS* der Firma SPACIAL [192], *OpenCASCADE* von OPEN CASCADE S.A.S. [168] oder *Parasolid* von EDS [64] bieten teilweise über einen hybriden Ansatz ähnliche Funktionalitäten im Umgang mit Oberflächenmodellen.

Uniformes Normzellen-Aufzählungsschema. Bei der Beschreibung eines Volumenkörpers über das Normzellen-Aufzählungsschema wird der geometrische Raum als Gitter aufgeteilt, welches aus dreidimensionalen Zellen gleicher Kantenlänge, sogenannten *Voxeln* besteht (siehe Abbildung 6.5).

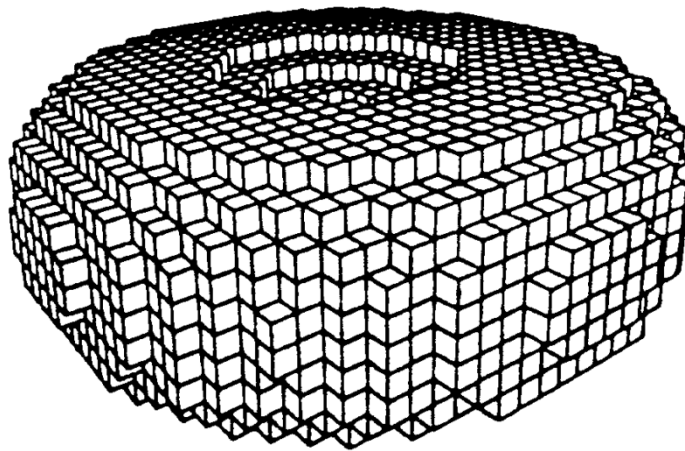


Abbildung 6.5: Darstellung eines geometrischen Körpers über das Normzellen-Aufzählungsschema als Gitter aus Würfeln gleicher Kantenlänge.

Jede Zelle wird über ihre Mittelpunktkoordinate definiert und ist abhängig von der Position des Mittelpunktes (innerhalb oder außerhalb eines Körpers) eindeutig bestimmt. Ein Voxel kann bei einer Definition über eine *Bit-Matrix* den Wert 0 (für *Fluidknoten*) oder 1 (für *Solidknoten*) annehmen.

Um einen Knotenpunkt mit zusätzlichen Attributen, wie beispielsweise Simulationsrandbedingungen zu verknüpfen, kann der repräsentierende *Bit*-Datentyp durch den Datentyp *Integer* ersetzt werden. Hierdurch ist ein eindeutiger Verweis auf Objekte einer Attributliste über die Datenstruktur gegeben.

Die Genauigkeit eines uniformen Gitters ist abhängig von der Kantenlänge der Voxel und benötigt bei einem hochauflösenden Raster einen hohen Speicherplatzbedarf. Im Falle eines dreidimensionalen Körpers liegt die Komplexität eines uniformen Normzellen-Aufzählungsschemas bei $O(n^3)$ [39].

Oktalbaum (Octree). Das im Bereich der Bildrepräsentation häufig eingesetzte und in der Abbildung 6.6 dargestellte Octree-Schema (im zweidimensionalen Fall auch als Quadtree-Schema [179] bezeichnet) bietet einen reduzierten Speicherbedarf im Vergleich zum uniformen Normzellen-Aufzählungsschema bei gleicher Genauigkeit der Modellbeschreibung.

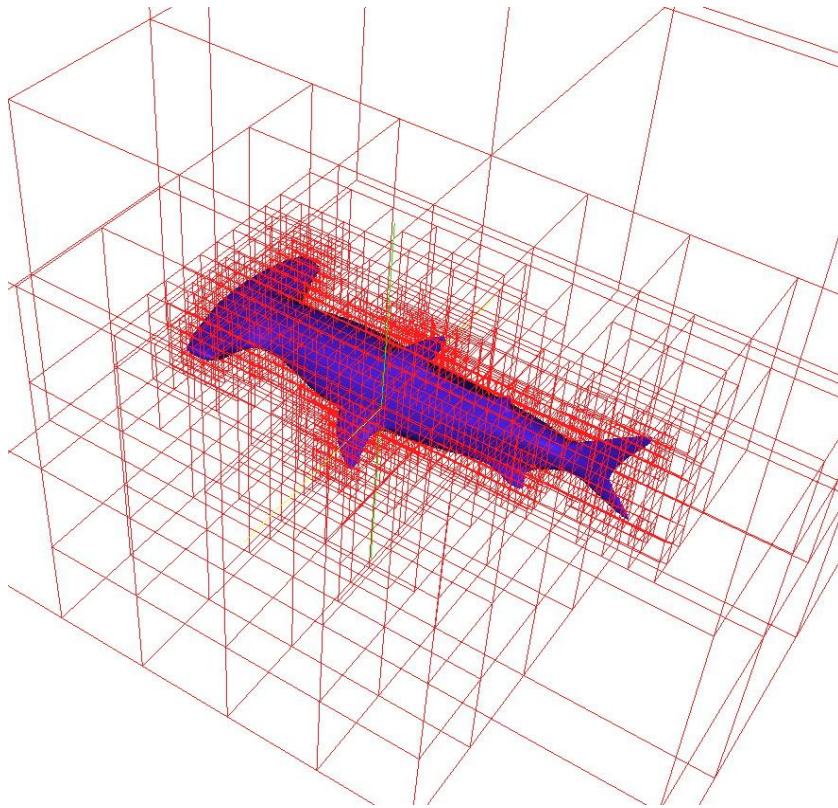


Abbildung 6.6: Modellierung eines Objektes mit gekrümmten Oberflächen mittels Octrees

Basis für den Diskretisierungsprozess bietet hierbei die *Bounding-Box* (siehe Abschnitt 3.3.2 und Abschnitt 3.3.3) welche das Berechnungsgebiet in Form eines Würfels vollständig umschließt. Dieser Würfel wird rekursiv in Teilwürfel halber Kantenlänge unterteilt (acht Würfel für den dreidimensionalen Fall und vier Würfel für den zweidimensionalen Fall) bis ein Abbruchkriterium erfüllt ist. Hierbei kann es sich um das Erreichen einer vorgegebenen Genauigkeit für die kleinste Kantenlänge eines Würfels handeln (als *Tiefe* oder *Level* bezeichnet) oder die eindeutige Bestimmtheit des Würfels. Ein Würfel ist eindeutig bestimmt, sobald er vollständig innerhalb oder außerhalb des darzustellenden Körpers liegt. Der Würfel kann somit die Zustände „in“ (innerhalb), „off“ (außerhalb) oder „on“ (schneidet den Rand) annehmen.

Das Datenschema stellt eine Baumstruktur dar, welche eine vierfache (zweidimensional) bzw. achtfache (dreidimensional) Verästelung aufweist (siehe Abbildung 6.7).

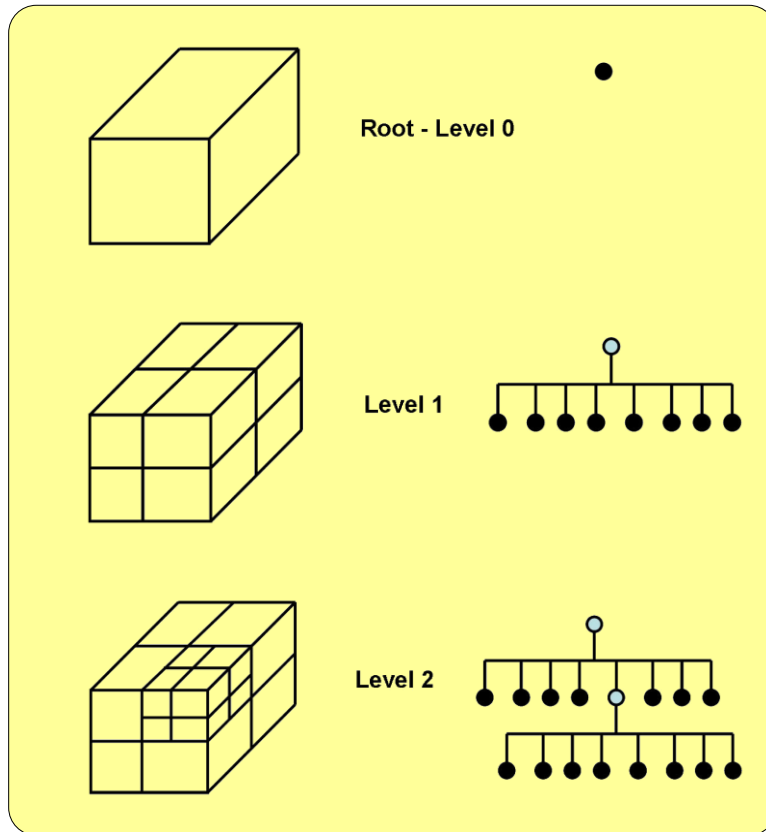


Abbildung 6.7: Die Datenstruktur eines Oktalbaums

Die unterteilten Würfel werden als *Knoten* bezeichnet, Würfel ohne weitere Unterteilung als *Blätter*. Jeder Knoten, mit Ausnahme des *Root-Oktanten* (der Basis-Würfel, welcher das gesamte Berechnungsgebiet umschließt) verweist auf genau einen Vorgänger. Der Level oder die Tiefe eines Baumes wird über den Abstand des untersten Blattes zum Root-Oktanten bestimmt. Eine spezielle Ausprägung des Oktalbaums bilden die *geglätteten Bäume*, bei denen benachbarte Würfel einen maximalen geometrischen Größenunterschied über eine definierte Grenze hinaus nicht überschreiten [50].

Der Oktalbaum zeigt in Kombination mit der Nutzung von Facettenmodellen eine effiziente Möglichkeit der Gittergenerierung [114]. Der Speicherplatzbedarf dieses Schemas reduziert sich gegenüber dem Normzellen-Aufzählungsschema um eine Größenordnung. Die Komplexität liegt bei $O(n^{D-1})$, wobei D die Raumdimension angibt. Für ein dreidimensionales Modell besitzt der Oktalbaum somit eine Mächtigkeit von $O(n^2)$ gegenüber einer Mächtigkeit von $O(n^3)$ im Falle der uniformen Darstellung [39].

6.3 Überblick

Abbildung 6.8 und Abbildung 6.9 veranschaulichen den automatisierten Modelltransfer über den Software-Prototypen. Über die Konvertierung eines IFC Produktmodells zu einem Berechnungsmodell bis zur Visualisierung ist eine Vielzahl komplexer Teilschritte zu durchlaufen, welche über die Grafiken in fünf Hauptkategorien unterteilt werden:

- **Objekt-Attribute:** Diese Kategorie stellt die Datenstruktur der Randbedingungen für Bauteilelemente dar, welche für die numerische Simulation benötigt werden.
- **Schnittstellen:** Das Durchlaufen des Modelltransfers erfolgt über mehrere Einzelapplikationen. Die für die Kommunikation zwischen den Instanzen und über die Infrastruktur des Rechnerverbundes notwendigen Schnittstellen werden in dieser Kategorie dargestellt.
- **Datenbasis:** Die Konvertierung der Geometrie stellt eine der Hauptaufgaben des Modelltransfers dar. Die Abbildung dieser geometrischen Informationen erfolgt über die im Abschnitt 6.2 vorgestellten Modellierungsstrukturen.
- **Software-Applikation:** Der Prototyp besteht aus einem Verbund automatisierter Software-Module. Die Aufgaben dieser Applikationen für den Modelltransfer werden hier dargestellt.
- **Hardware:** Neben einem Workstation-PC zur Modellierung und Visualisierung dient ein PC-Cluster zur Durchführung von Teilbereichen des Pre-Processing und der numerischen Simulation. Diese Kategorie gibt Aufschluss über die Hardware, auf der ein Prozess ausgeführt wird.

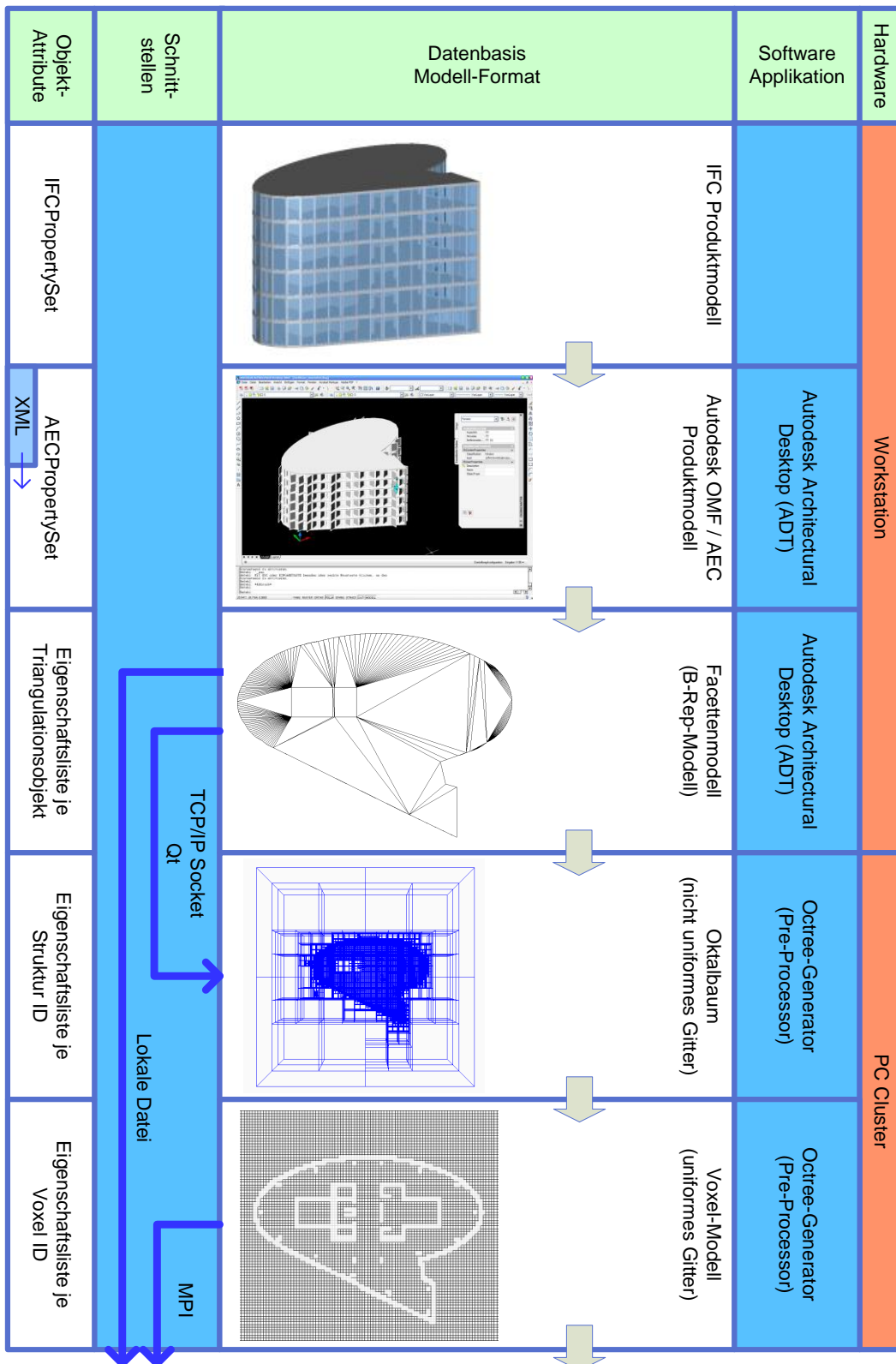


Abbildung 6.8: Automatisierter Modelltransfer innerhalb des Software-Prototyps, Teil 1

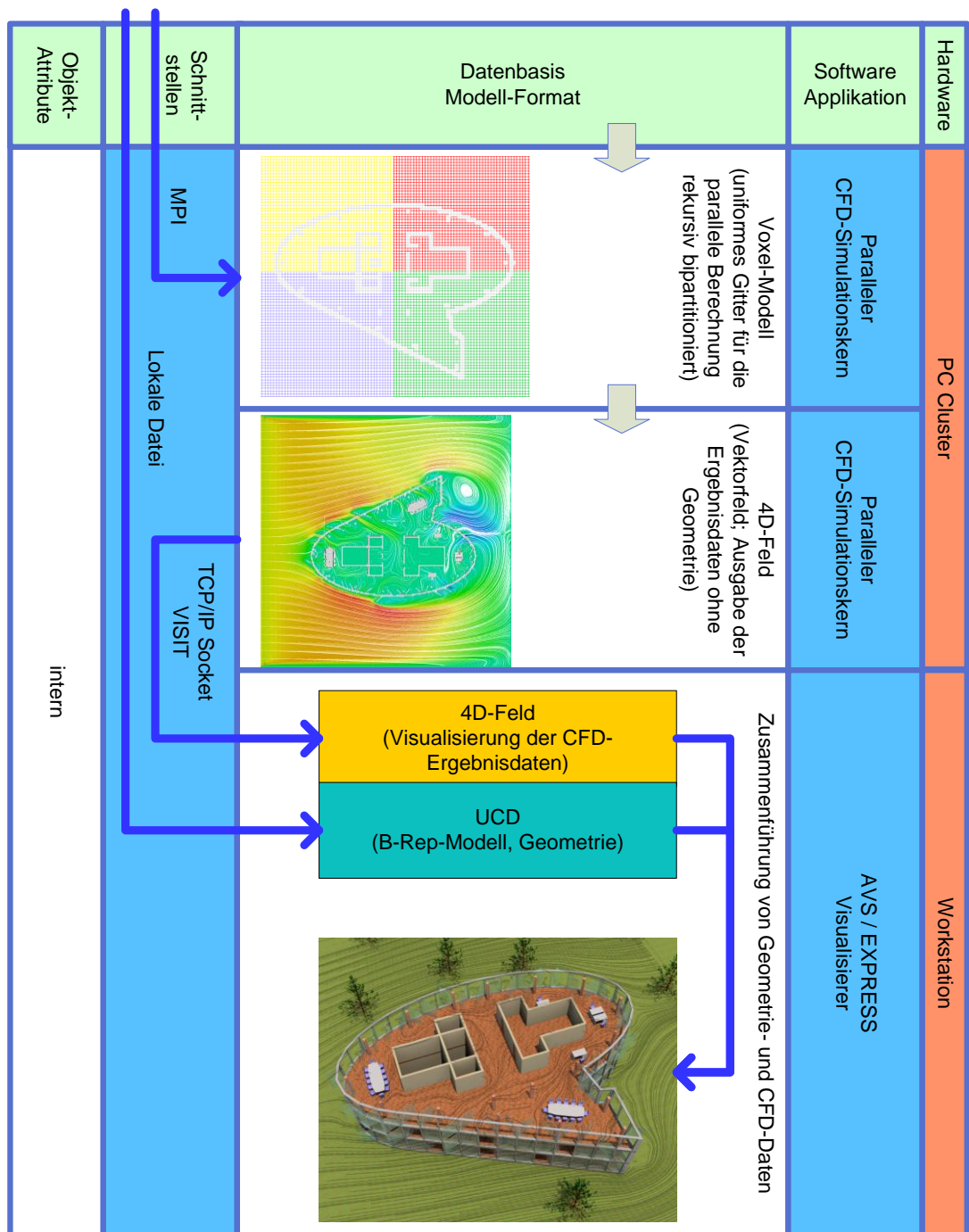


Abbildung 6.9: Automatisierter Modelltransfer innerhalb des Software-Prototyps, Teil 2

6.4 Schritt I: Vom IFC Modell zum AEC Modell

Der folgende Abschnitt beschreibt die erste Instanz des Modelltransfers, den Import des IFC Produktmodells und die Erweiterung der Bauteile mit numerischen Randbedingungen.

6.4.1 Konvertierung der IFC-Geometrie

Die Volumengeometrie der IFC-Objekte, definiert über die abstrakte Klasse *IfcSolidModel*, kann über verschiedene geometrische Modelle beschrieben werden. Häufige Anwendung finden hierbei folgende Ansätze:

- Darstellung der Außenfläche als *facettierte Oberflächenmodell* (*Faceted B-rep*) über die Struktur dreidimensionaler Polyeder [63].
- Die *Extrusion* einer ebenen Fläche entlang einer vorgegebenen Extrusions-Kurve.
- Besondere Bauteile wie beispielsweise Wände können über eine *parametrisierte Definition* [23] des Wand-Volumens aus Wand-Achse, Dicke und Höhe über Volumenkörper erzeugt werden.

Weitere Darstellungsmöglichkeiten finden sich über das *Sweeping-Modell* oder das *CSG-Modell*. Durch Nutzung des ADT Plug-In *IFC-Utility 2x for ADT 2005* [80] können AEC-Repräsentanten der IFC-Objekte auf dem Produktmodell für das ADT erzeugt werden. Die geometrischen Informationen werden in den Modellierer überführt und als Attribute an die AEC-Objekte angehängen.

Der AModeler. Die Verarbeitung der Objektinformationen wird hierbei vom *AModeler* [19], dem Geometriekern des Architectural Desktops durchgeführt, welcher einen großen Funktionsumfang zur Bearbeitung geometrischer Daten besitzt:

- Optimierte Widged-Edge/B-rep Datenstruktur zur facettierten Beschreibung von Oberflächen,
- Modellierung von manifold und non-manifold Körpern [39],
- boolesche Mengenoperationen für Körper (Durchschnitt, Vereinigung, Differenz) (siehe CSG-Modell),
- Modellierung von Grundprimitiven (beispielsweise Zylinder, Kugel, Würfel),
- Schnittführung durch Körper,
- Sweeping-Operationen,
- Extrusion,
- geometrische Transformationen (beispielsweise Translation, Rotation, Skalierung, Spiegelung)
- Berechnung von geometrischen Eigenschaften eines Körpers wie beispielsweise Oberfläche, Volumen und Schwerpunkt,
- Prüfung auf Überschneidungen von Körpern,
- Punkt in Polyeder Test,
- Bestimmung der konvexen Hülle für eine Punktmenge im Raum.

Die Abbildung 6.10 stellt den schematischen Aufbau der Topologie des in den AModeler implementierten B-rep Modells dar.

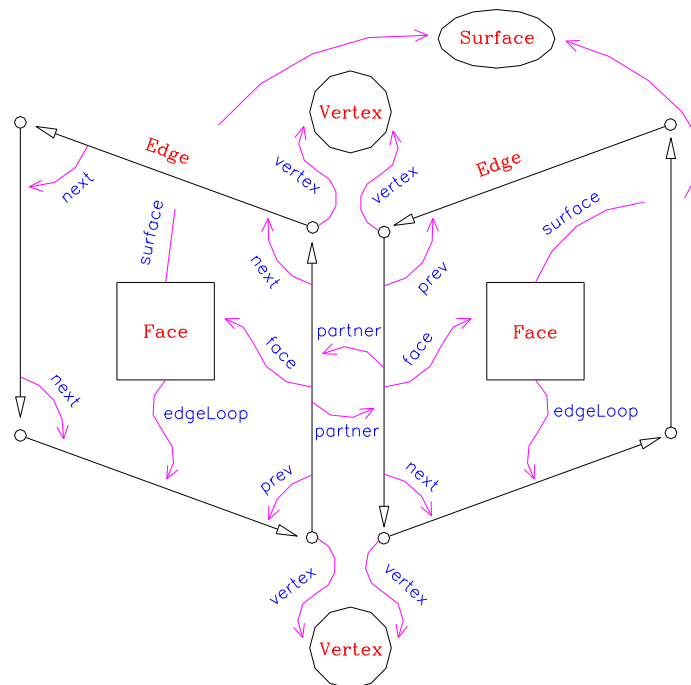


Abbildung 6.10: Topologische Struktur des geometrischen Kerns AModeler

Die Oberfläche des geometrischen Körpers wird über die Objekte *Body*, *Surface*, *Face*, *Edge* und *Vertex* begrenzt. Der Normalenvektor zeigt in Richtung der Außenseite des Körpers. Über den AModeler erzeugte Flächen können nicht über Krümmungen beschrieben werden. Hierzu werden Approximationskanten (siehe Abbildung 6.11) eingeführt, welche die Oberflächenstruktur in *Facetten* unterteilen.

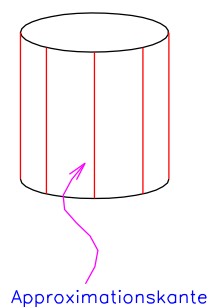


Abbildung 6.11: Unterteilung der Körperoberfläche in Facetten durch Approximationskanten

Die erzeugten Facettenelemente (*Faces*) verweisen auf ihre abgeleiteten Oberflächen (*Surface*). Eine geschlossene Folge (*loop*) von Kanten (*edges*) beschreibt ein Facettenelement.

6.4.2 Konvertierung bauteilspezifischer Metainformationen

Bauteilinformationen als Eigenschaftssätze. Neben den geometrischen Daten der IFC-Objekte werden auch deren bauteilspezifische Metainformationen verlustfrei in das AEC Produktmodell überführt. Das Plug-In *IFC-Utility 2x for ADT 2005* der Firma INOPSO [80] erlaubt die konsistente Konvertierung der Metainformationen von IFC Eigenschaftssätzen (*IfcPropertysets* [94, 97]) in Eigenschaftssätze des Autodesk Produktmodells (*AEC Property Sets* [15, 17]). Abbildung 6.12 beschreibt den schematischen Aufbau eines *IfcPropertysets* über die *Express-G* Notation.

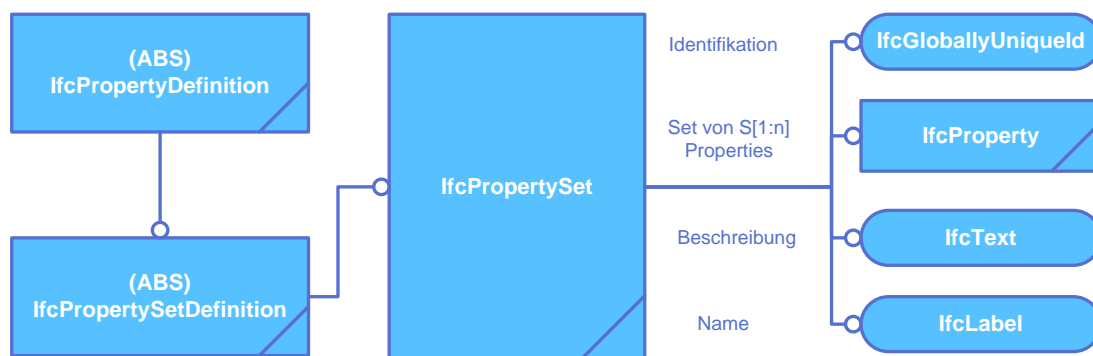


Abbildung 6.12: Aufbau eines *IfcPropertySet*s

Ein *IfcPropertySet* kann über eine eindeutige Objekt-ID, die *IfcGloballyUniqueId*, zugeordnet werden [167]. Diese ID behält über die gesamte Lebensdauer eines Bauteilelementes ihre Gültigkeit. Neben Namensgebung und Beschreibungstext kann ein Eigenschaftssatz mit einer beliebigen Anzahl von Attributen (*IfcProperty*) erweitert werden.

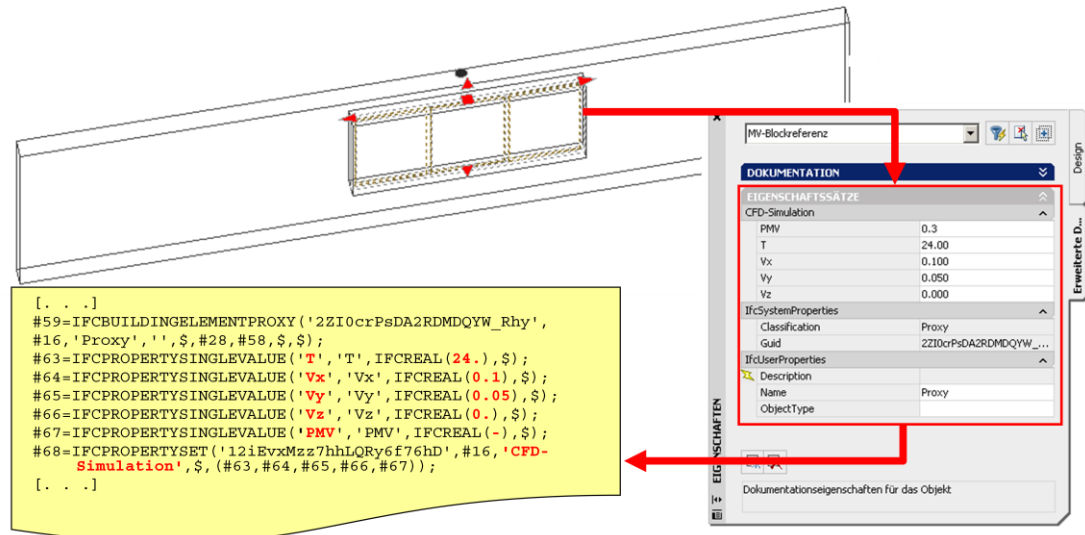


Abbildung 6.13: Konsistente Überführung der Metainformationen von IFC Bauteilelementen in Objekteigenschaften des AEC Produktmodells

Die Abbildung 6.13 zeigt beispielhaft eine Auswahl von Attributen eines Klimaanlageauslasses für die CFD-Simulation. Der linke Bereich der Abbildung stellt den Quelltext des IFC Produktmodells für dieses Objekt dar. Hierbei sind die Objektinformationen als Einzelwerte (*IfcPropertySingleValue*) des *IfcPropertySets* „CFD-Simulation“ gespeichert. Der rechte Teil der Abbildung stellt analog dazu die Metainformationen des überführten Objektes über das AEC Produktmodell als *AEC Property Sets* dar. IFC-spezifische Informationen des Modells, wie IFC Klassenzugehörigkeit, GUID, Beschreibung und Namensgebung werden über weitere Eigenschaftssätze mit dem AEC Objekt verknüpft. Eine konsistente Rückführung des AEC Modells in ein IFC Produktmodell ist somit gewährleistet.

Randbedingungen für die Simulation. Das IFC Produktmodell unterstützt keine standardisierte Implementierung CFD-spezifischer Randbedingungen sowie weitere, für den interaktiven Prototyp notwendige Parameter und Objektattribute. Die geforderten Eigenschaften lassen sich im Folgenden darstellen:

- Attribute für die numerische Simulation, beispielsweise Randbedingungen für Temperatur, Geschwindigkeit, Druck, Strahlung und Materialeigenschaften.
- Attribute für die Visualisierung der Geometrie, beispielsweise Farbe und Transparenz.
- Attribute für das HVAC-Agenten Framework, beispielsweise Parameter für Reglerglieder und Grenzwerte.
- Attribute für die Überwachungswerkzeuge des Konstruktions- und Entwurfsprozesses, wie beispielsweise Grenzwerte für Eingabeparameter von Randbedingungen, HVAC Parameter oder Ankerelemente.
- Attribute für die fachplanerspezifische Zuordnung von Objekten für die automatische Erstellung von Bauplänen.

Das Datenmodell wird mit den geforderten Objektattributen über Eigenschaftssätze (Property Sets) erweitert und mit den Bauteilelementen verknüpft. Ein Prozess überprüft während des Ladevorganges für ein IFC- oder AEC- Produktmodell in den ADT, ob notwendige Eigenschaftssätze existieren³³. Hierbei validiert der in der Abbildung 6.14 dargestellte Prozess über eine ADT Datenbank alle Bauteile einer Konstruktionszeichnung und überprüft die Eigenschaftssätze jedes Objektes.

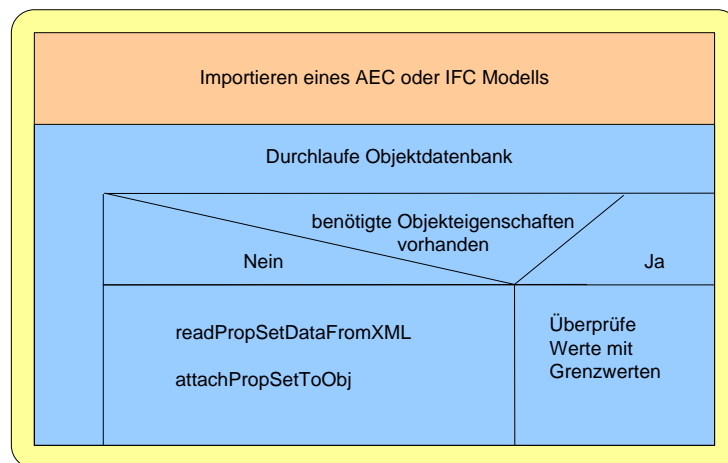


Abbildung 6.14: Funktionsweise der Importroutine für die Modellaufbereitung innerhalb des ADT

Existieren benötigte Objektattribute und Randbedingungen, werden die Werte dieser Eigenschaftssätze auf ihre Richtigkeit überprüft. Dies erfolgt über einen Vergleich mit den geforderten Minimal- und Maximalwerten. Bei einem Fehlen der Randbedingungen werden in einer *XML-Datei* [218] hinterlegte Datensätze objektspezifisch an das Bauteilelement angehängen. Kann ein Objekt nicht eindeutig zugeordnet werden, wird ein Datensatz mit allgemeingültigen Randbedingungen verwendet.

Der nachfolgende XML-Auszug beschreibt die Vorlage für einen Eigenschaftssatz, welcher die notwendigen Simulationsrandbedingungen vorhält. Es werden Name, Wert und Datentyp des jeweiligen Objektattributs deklariert. Der hier dargestellte Datensatz enthält Randbedingungen mit Standardwerten für nicht zuordnungsfähige Objekte.

³³ Die als Eigenschaftssätze gespeicherten Attribute werden für einen Datenaustausch zwischen den beteiligten Fachplanern konsistent in ein IFC-Modell überführt und müssen bei wiederholtem Einlesen nicht neu erstellt werden.

```

<PropertySetDefinitions>
  <propset name="simulation">
    <attribute name="type"    defaultValue="default" dataType="string"/>
    <attribute name="type_ID" defaultValue="1"      dataType="int"    />
    <attribute name="ACAD_ID" defaultValue=" "      dataType="int"    />
    <attribute name="T"        defaultValue="0.0"    dataType="double" />
    <attribute name="u_x"      defaultValue="0.0"    dataType="double" />
    <attribute name="u_y"      defaultValue="0.0"    dataType="double" />
    <attribute name="u_z"      defaultValue="0.0"    dataType="double" />
    <attribute name="P"        defaultValue="0.0"    dataType="double" />
    <attribute name="H"        defaultValue="0.5"    dataType="double" />
    <attribute name="RAD_rho"  defaultValue="0.8"    dataType="double" />
    <attribute name="RAD_tau"  defaultValue="0.0"    dataType="double" />
    <attribute name="noslip"   defaultValue="1"      dataType="int"    />
  </propset>
</PropertySetDefinitions>

```

Eigenschaftssätze für spezifische Bauteilobjekte, in diesem Beispiel für eine Wand und einen Klimaanlageinlass, sind im Folgenden dargestellt:

```

<SimObjects>
  <object name="AecDbWall">
    <attribute name="T"          value="20"          dataType="double"/>
    <attribute name="u_x"        value="0.0"          dataType="double"/>
    <attribute name="u_y"        value="0.0"          dataType="double"/>
    <attribute name="u_z"        value="0.0"          dataType="double"/>
    <attribute name="type_ID"    value="1010"         dataType="int"    />
    <attribute name="P"          value="0.0"          dataType="double"/>
    <attribute name="H"          value="0.5"          dataType="double"/>
    <attribute name="RAD_rho"    value="0.8"          dataType="double"/>
    <attribute name="RAD_tau"    value="0.0"          dataType="double"/>
    <attribute name="noslip"     value="1"            dataType="int"    />
  </object>

  <object name="HVACWallInlet1">
    <attribute name="T"          value="22.0"         dataType="double"/>
    <attribute name="u_x"        value="0.3"          dataType="double"/>
    <attribute name="u_y"        value="0.0"          dataType="double"/>
    <attribute name="u_z"        value="0.0"          dataType="double"/>
    <attribute name="type_ID"    value="6110"         dataType="int"    />
    <attribute name="P"          value="0.0"          dataType="double"/>
    <attribute name="H"          value="0.5"          dataType="double"/>
    <attribute name="RAD_rho"    value="0.8"          dataType="double"/>
    <attribute name="RAD_tau"    value="0.0"          dataType="double"/>
    <attribute name="noslip"     value="1"            dataType="int"    />
  </object>
</SimObjects>

```

Dieser Ansatz bietet die Möglichkeit einer dynamischen Erweiterung von Eigenschaftssätzen für Bauteiltypen und ihrer Attribute. Dieses erfolgt über die Modifizierung der XML-Datei. Änderungen werden direkt von der ADT-Applikation

übernommen. Die Funktionsweisen zur Verarbeitung von Eigenschaftssätzen sowie XML-Daten werden in Abschnitt 3.3 beschrieben.

6.5 Schritt II: Vom AEC Modell zum Facettenmodell

Die Prozessinitiierung für die automatische Konvertierung der Geometrie des AEC Produktmodells in ein Facettenmodell erfolgt über die in Abschnitt 3.3.3 beschriebenen Ereignisprotokolle.

Die geometrischen Attribute der als AEC Objekte (*AEC Entity*) vorliegenden Bauteile werden als *Solids* des in Abschnitt 6.4.1 beschriebenen *Amodelers* vorgehalten. Das OMF bietet keine Methoden für eine Manipulation dieser Attributdaten, unterstützt aber über einen *Streaming*-Mechanismus die schnelle Extraktion der Solid-Information aus einem AEC Entity. Die Voraussetzung bilden hierbei die Klassen *AecStreamAcad* und *AecStreamCollectBodies* [15].

Die gewonnenen Geometriedaten (*Bodies*) werden nachfolgend über die vom AModeler zur Verfügung gestellten Methoden³⁴ *trianguliert*. Die in der Abbildung 6.15 dargestellte Triangulation umfasst die Unterteilung in Dreiecksfacetten, welche die Oberfläche des Körpers beschreiben.

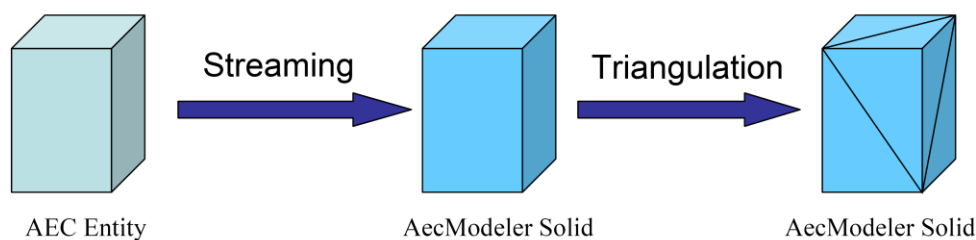


Abbildung 6.15: Die Triangulation beschreibt die Extraktion der geometrischen Informationen aus AEC Objekten und die Generierung einer Oberflächenstruktur basierend auf Dreiecksfacetten.

Das hieraus resultierende Facettenmodell, wie beispielsweise in der Abbildung 6.16 dargestellt, findet in der Visualisierungstechnik häufige Anwendung. Im Bereich des Bauwesens stellen die Einschränkungen der Modellstruktur in Bezug auf gekrümmte Oberflächen keine wesentlichen Nachteile dar, da Bauteile vorwiegend plane Oberflächen aufweisen.

³⁴ Der AModeler basiert auf einer objektorientierten Softwarestruktur und bietet über eine C++ Schnittstelle, welche über das OMF angesprochen werden kann, Zugriff auf seine geometrischen Funktionen.

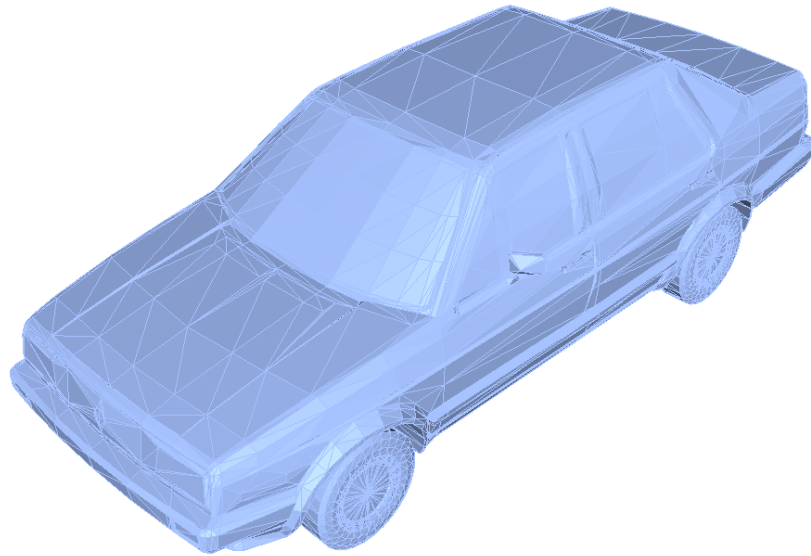


Abbildung 6.16: Darstellung des Oberflächenmodells für ein Fahrzeug durch Dreiecksfacetten (Facettenmodell)

Die Raumkoordinaten aller Dreieckselemente eines Objektes werden iterativ in eine temporäre Liste für den Sendeprozess übertragen. Die Datenextraktion erfolgt über das Auslesen der Oberflächeninformationen des Facettenmodells. Die Liste der Koordinaten wird selektiv pro Objekt erstellt, um notwendige Randbedingungen für die Simulation erweitert und mit einer eindeutigen Objekt-Identifikationsnummer markiert. Für eine detaillierte Beschreibung des Sendeprozesses wird auf Abschnitt 3.3.3.1 und 3.3.3.2 sowie auf [219] verwiesen.

Die geometrischen Dreiecksinformationen werden parallel für die Generierung einer Datei im *UCD Format* (*Unstructured Cell Data*; [5]) zur anschließenden grafischen Ausgabe über das Programm *AVS/EXPRESS* [6] genutzt. Durch die Visualisierung der Geometrie über Dreieckselemente wird ein höherer Detaillierungsgrad gegenüber der Geometriedarstellung über Knotenpunkte erzielt. Die über den Simulationskern übertragene geometrische Struktur von Objekten basiert auf Knotenpunkten.

6.6 Schritt III: Vom Facettenmodell zum Oktalbaum

Das Facettenmodell sowie die Randbedingungen für die numerische Simulation werden über eine automatisierte Kommunikationsschnittstelle an den PC-Cluster übertragen. Der Empfang erfolgt über den *Pre-Processor*. Eingesetzt wird hierbei ein auf einem *Oktalbaum* (siehe [37, 114] und Abschnitt 6.2) basierender Gittergenerator. Seine Struktur beinhaltet Optimierungsansätze zur effizienten Generierung hochauflösender Oktalbäume [219]. Der Gittergenerator wurde mit der Programmiersprache C++ [108] entwickelt.

Über den in Abschnitt 6.2 vorgestellten Oktalbaum werden Objektoberflächen über Würfel repräsentiert. Hierbei findet, ausgehend von einem das geometrische Objekt einhüllenden Würfel, eine Unterteilung in Würfel halber Kantenlänge statt bis ein

Abbruchkriterium erfüllt ist. Dieses kann über das Erreichen eines maximalen Levels (Auflösungsgrad) oder eines eindeutigen Zustandes des Würfels erfolgen. Eine Verfeinerung des Würfels findet nur statt, wenn diese mit den Facetten des geometrischen Modells eine nichtleere Schnittmenge bilden und somit nicht eindeutig bestimmt sind. Das Verfahren zur Bestimmung der Schnittmenge erfolgt über den *triangle-box overlap Test*.

6.6.1 Der triangle-box overlap Test

Der *triangle-box overlap Test* von AKENINE-MÖLLER [7] beschreibt eine effiziente Routine zur Feststellung von Überlappungen zwischen Dreieck und Würfel im dreidimensionalen Raum. Das zu Grunde liegende Verfahren basiert auf dem *Separating Axis Theorem* [49] aus dem Bereich der Computergrafik, im Speziellen des *Raytracing*.

Ein Schnittmengentest zwischen Objekten ist in der Regel teuer, da Objekte aus vielen Polygonen bestehen können. Zur Beschleunigung dieses Testverfahrens bedient man sich einer einfachen Methode. Hierbei werden Hüllkörper, so genannte *Bounding Volumes*, mit einer einfachen Geometrie um Objekte gelegt. Ein Beispiel für einen Hüllkörper ist die *Sphäre*, die *Axis Aligned Bounding Box (AABB)*, die *Oriented Bounding Box (OBB)* und die *Discrete Oriented Polytopes (k-Dop)* [45].

Eine *Axis Aligned Bounding Box (AABB)* ist ein Quader, bei dem die Oberflächennormalen mit den Achsen der Standardbasis übereinstimmen. Der Schnittstellentest zwischen zwei AABBs erfolgt über eine Dimensionsreduzierung mit einer nachfolgenden Kontrolle, ob sich die Projektionen der beiden Quader auf den X-, Y- und Z-Koordinatenachsen des Basissystems überschneiden. Ist dies der Fall, schneiden sich auch die AABBs. Gibt es mindestens eine Koordinatenachse ohne einen Schnitt der Projektionen, schneiden sich auch keine AABBs [45].

Eine Verbesserung des AABB Hüllkörpers stellt die *Oriented Bounding Box (OBB)* dar, deren Ausrichtung sich nicht mehr an der Standardbasis orientiert, sondern eine möglichst enge Passform um das umschließende Objekt zu Verbesserung der Kollisionserkennung aufweist. Bei einer Dimensionsreduzierung und einem Schnittmengenvergleich der Objektprojektionen auf den Koordinatenachsen des Basissystems (analog dem Schnittstellentest für AABBs) existiert die Möglichkeit, dass sich die OBBs nicht schneiden, obwohl eine Schnittmenge einer oder mehrerer Projektionen der beiden Körper existiert. Eine Lösung stellt die Einführung einer in der Abbildung 6.17 dargestellten *Trennachse (Separating Axis)* dar.

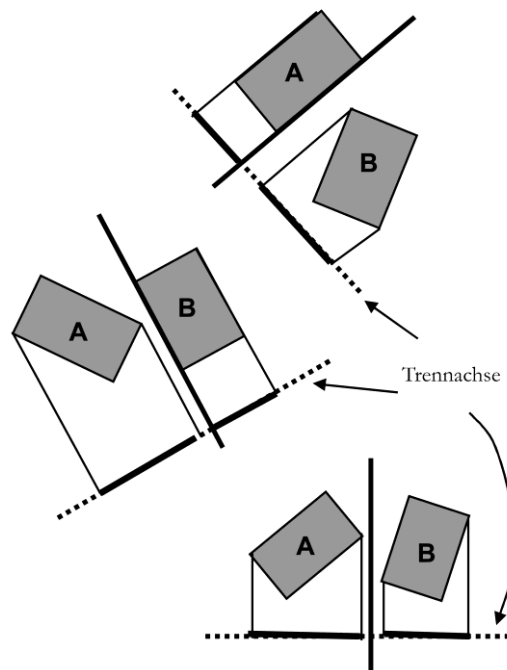


Abbildung 6.17: Separating Axis Theorem

Die beiden dargestellten Objekte A und B schneiden sich nicht, wenn eine eingeführte Trennachse existiert, also die Projektionen dieser Objekte auf der Trennachse keine Schnittmenge bilden. Sind die beiden Objekte disjunkt, existiert eine Trennachse. Diese ist entweder zu einer Seitenfläche von A, einer Seitenfläche von B oder jeweils einer Kante von A und B orthogonal. Für eine ausführliche Beschreibung sowie Spezialfälle des Separating Axis Theorems wird auf GOTTSCHALK [84] verwiesen.

Über das hier vorgestellte Verfahren werden mit dem hier eingesetzten Gittergenerator Tests zur Bestimmung einer möglichen Trennachse und somit einer existierenden Schnittmenge durchgeführt [219].

6.6.2 Intelligentes Lokalisierungsverfahren

Die Optimierungsansätze für den hier vorgestellten Oktalbaum hinsichtlich der Geschwindigkeitserhöhung bei der Erzeugung hochauflösender Gitter basiert auf einer *Komplexitätsreduzierung des Lokalisierungsverfahrens*. Hierbei werden die geometrischen Informationen (Dreiecksnetze) in einer zusätzlichen Baumstruktur für die jeweiligen Oktalbaum-Oktanten vorgehalten.

Die Schnittmengenbestimmung eines Oktalbaums erfolgt mit dem im Vorangegangenen erläuterten *triangle-box overlap Test*. Ein herkömmliches Lokalisierungsverfahren untersucht hierbei für jeden Würfel *alle* Dreiecke des Facettenmodells und erreicht damit eine Komplexität von $O(n^3)$. Gerade bei komplexen Oberflächenmodellen mit einer hohen Anzahl von Dreieckselementen führt dies zu Geschwindigkeitsreduzierungen infolge

unnötiger Untersuchungen von Dreieckselementen außerhalb des betrachteten Oktalbaum-Quadranten.

Der hier vorgestellte Lokalisierungsansatz basiert auf der Umsetzung der Strategie *teile und herrsche*³⁵ aus dem Bereich der Informatik. Hierbei werden die Dreiecke des Facettenmodells in einer zusätzlichen Baumdatenstruktur verwaltet. Bei der Teilung der Geometrie eines Würfels in einzelne Quadranten über den Oktalbaum erfolgt eine Zuordnung der Dreieckselemente für die jeweiligen Oktalbaum-Quadranten. Die Baumstruktur erlaubt hierbei die Generierung einzelner *Unterbäume*, welche die Dreieckselemente für einen zugeordneten Quadranten beinhalten. Bei einer Diskretisierung werden zur Bestimmung der Schnittmenge zwischen Facette und Würfel nunmehr Dreieckselemente der Unterbäume bzw. Quadranten überprüft. Die Untersuchung des Gesamtmodells entfällt.

Abbildung 6.18 erläutert das Prinzip des eingesetzten Lokalisierungsverfahrens über das zweidimensionale Beispiel eines Quadtree.

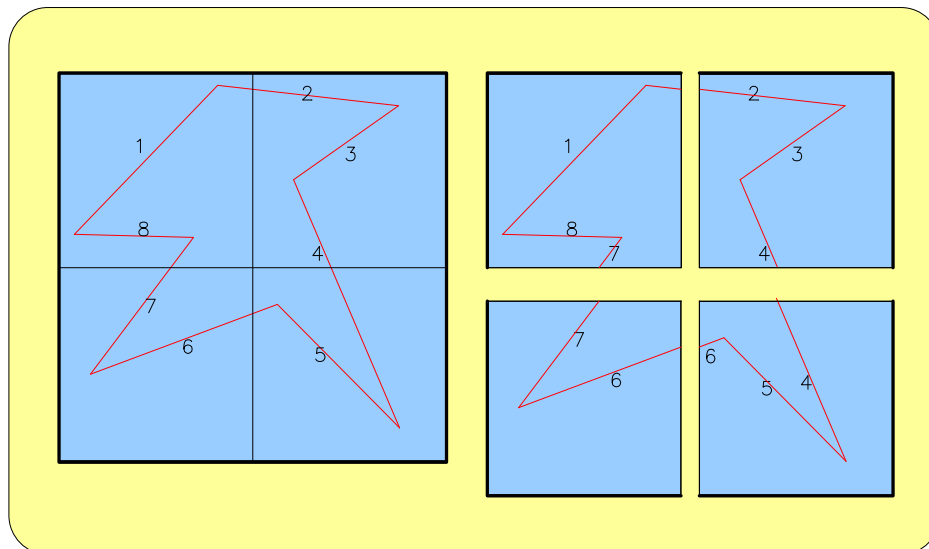


Abbildung 6.18: Darstellung der Suchstrategie des angewendeten Lokalisierungsverfahrens für einen zweidimensionalen Quadtree

Das dargestellte Polygon besitzt acht Liniensegmente. Nach der Diskretisierung erfolgt die Aufteilung des Root-Quadranten in vier Unterquadranten. Mit diesem Vorgang werden die über die Baumdatenstruktur des Polygons vorgehaltenen Liniensegmente in Unterbäume übertragen und den entsprechenden Quadranten zugeordnet. Bei einer weiteren Diskretisierung des in der Abbildung 6.18 oben rechts dargestellten Quadranten überprüft der Lokalisierungsalgorithmus nur die Liniensegmente 2, 3 und 4. Die Überprüfung anderer Liniensegmente ist nicht erforderlich. Bei jedem Diskretisierungsschritt eines Quadtree werden vier Unterbäume erzeugt, für einen Octree erfolgt die Erzeugung von acht Unterbäumen. Je weiter die Diskretisierung voranschreitet, desto

³⁵ Der Grundsatz der Informatik „teile und herrsche“ (engl. „divide and conquer“ oder lat. „divide et impera“) beschreibt einen reduktionistischen Lösungsansatz. Das Prinzip basiert auf der Reduzierung des Lösungsaufwandes durch die Zerlegung des Problems in kleinere Teilprobleme. Bei Optimierungsproblemen wird der Lösungsraum aufgeteilt und die Lösungssuche auf die Unterräume verlagert.

kleiner werden die Oktalbaum-Würfel und desto geringer wird die Anzahl der zu überprüfender Segmente.

Der hier nachfolgende Quellcode beschreibt den Aufbau des angewendeten Algorithmus zur Erzeugung von Unterbäumen. Der Prozess wird hierbei über den *triangle-box overlap Test* initiiert. Die übergeordnete Baumstruktur wird durch die Funktion `GetNewTree` übergeben.

```

procedure GetNewTree(Oct, ParentTree)
    for all Triangle facets T of ParentTree do
        Overlap  $\leftarrow$  Triangle-Box Overlap(Oct, ParentTree)
        if Overlap then
            SubTree  $\leftarrow$  T (push T into SubTree)
        end if
    end for
    SubTree.ID  $\leftarrow$  ParentTree.ID
    return SubTree
end procedure

```

Der Schnittmengentest erfolgt iterativ über alle Dreieckselemente der übergeordneten Baumstruktur. Bei nicht leeren Schnittmengen werden die Dreieckssegmente in die Unterbaumstruktur übertragen. Für eine eindeutige Zuordnung der Dreiecke zur Verknüpfung mit den Randbedingungen für die numerische Simulation wird die Objekt-ID an den Unterbaum übertragen.

Über diesen Ansatz kann die Komplexität eines herkömmlichen Lokalisierungsverfahrens von $O(n^3)$ auf eine Komplexität von $O(n \log n)$ reduziert werden.

6.6.3 Benchmarks

Der folgende Abschnitt stellt Geschwindigkeitsmessungen des Lokalisierungsverfahrens mit Unterbaumstrukturen dem Lokalisierungsverfahren ohne Optimierungsansätze gegenüber. Das in Abbildung 6.6 dargestellte Facettenmodell eines Hammerhais mit 21.894 Dreieckselementen dient als Testfall. Die Messungen wurden auf einem PC mit einer AMD Athlon XP 2500+ CPU, 1,5 GB Arbeitsspeicher und dem Betriebssystem MICROSOFT Windows XP durchgeführt. Verglichen werden mehrere Testfälle mit unterschiedlicher Auflösung des Oktalbaums.

Level	Anzahl Würfel	Zeit [sec] mit optimierter Lokalisierung	Zeit [sec] ohne optimierte Lokalisierung	Speicher [MB] mit optimierter Lokalisierung	Speicher [MB] ohne optimierte Lokalisierung
8	25.495	0,875	0,063	9,20	8,36
9	103.335	1,375	0,344	26,16	12,78
10	417.117	2,531	2,391	92,12	30,57
11	1.674.240	6,344	20,188	351,63	101,54
12	6.709.662	22,500	158,766	1200,00	378,21

Tabelle 6.1: Leistungsvergleich zwischen dem optimierten Lokalisierungsverfahren mit zusätzlichen Baumstrukturen und einem herkömmlichen Lokalisierungsverfahren.

Das Verfahren zeigt Nachteile hinsichtlich des Speicherbedarfs. Dabei verdreifacht sich im Allgemeinen die Speichernutzung durch die Einführung einer zusätzlichen Baumstruktur für Dreieckselemente. Das Verfahren zeigt keine Vorteile bei schwach aufgelösten Oktalbäumen, da die Erstellung der Unterbäume einen ähnlich hohen Zeitaufwand beansprucht wie die Lokalisierung von Dreieckselementen innerhalb des Gesamtmodells. Ab einer Anzahl von ca. 1 Million Würfel übersteigt hingegen die benötigte Zeit zur Gittergenerierung bei einem nicht optimierten Lokalisierungsverfahren den Zeitbedarf des optimierten Ansatzes um ein Vielfaches.

6.7 Schritt IV: Vom Oktalbaum zum Voxelmodell

Der Oktalbaum dient der Generierung einer *uniformen Voxelmatrix* als Grundlage des Berechnungsmodells für den numerischen Kern. Bei einer Voxelmatrix handelt es sich um ein in Abschnitt 6.2 beschriebenes *uniformes Normzellen-Aufzählungsschema*.

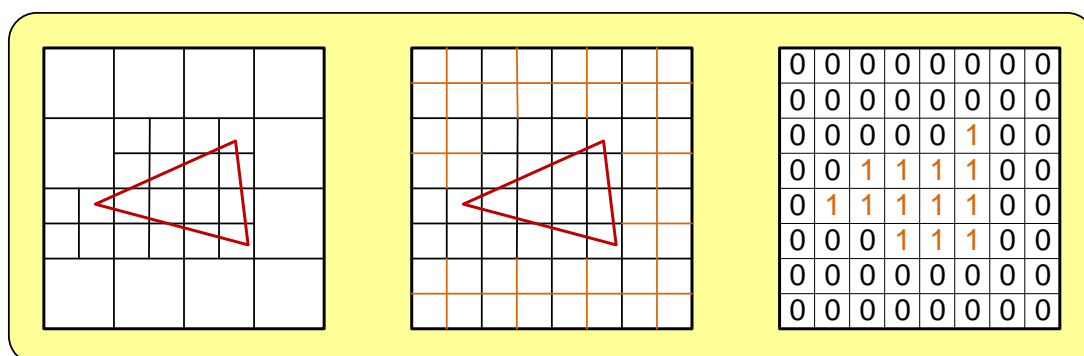


Abbildung 6.19: Konvertierung eines Oktalbaums in ein Voxelmodell über drei Teilschritte

Das dreidimensionale uniforme Gitter beschreibt den Raum, welcher über Würfel gleicher Kantenlänge gebildet wird. Die Bestimmung der Kantenlänge der Würfel ergibt sich über den kleinsten Knotenlevel des Oktalbaums (siehe Abbildung 6.19, linkes und mittleres Bild). Knoten des uniformen Gitters vom Datentyp *Integer* werden bei der Initialisierung der Voxelmatrix als Fluidknoten mit dem Wert „0“ deklariert. Die Knotenpunkte zur Markierung eines Körpers erhalten die in Abschnitt 6.4.2 erläuterte, eindeutige

Identifikationsnummer der Oktaalbaum-Repräsentanten. Dies ermöglicht die Zuordnung eines eindeutigen Satzes von Randbedingungen für jeden Knotenpunkt der Voxelmatrix (Abbildung 6.19, rechtes Bild).

Die Voxel des uniformen Gitters beschreiben die Oberfläche des Körpers. Jeder Körper besitzt somit einen Hohlraum. Zur Vermeidung unnötiger Berechnungen des Simulationskerns³⁶ muss das Innenvolumen aller Objekte in eine „massive“ Struktur überführt werden. Dies erfolgt über die Überführung von Voxeln, welche von der Oberfläche des Körpers umschlossen sind, in Solid-Voxel. Im Bereich der Informatik existieren mehrere effiziente Verfahren zum Füllen eines Körperinnenraumes. Lösungen liefern beispielsweise *rekursive Füll-Algorithmen* oder *Scan-Line-Algorithmen* [35, 39, 76]. Der im Rahmen dieser Arbeit eingesetzte Algorithmus basiert auf einem erweiterten, rekursiven Füll-Algorithmus [219]. Hierbei werden, ausgehend von einem beliebigen Knoten innerhalb eines Körpers, Nachbarknoten gesucht und mit der Identifikationsnummer dieses Körpers deklariert. Die Prozedur wiederholt sich über einen rekursiven Aufruf bis die Begrenzung des Körpers erreicht ist. Die Algorithmen variieren abhängig von der Anzahl der zu untersuchenden Nachbarknoten. Im zweidimensionalen Fall können vier Nachbarn (rechts, links, oben, unten) untersucht werden. Unter Berücksichtigung der diagonalen Nachbarschaften erhöht sich die Anzahl der zu untersuchenden Knoten auf acht. Im dreidimensionalen Raum liegt die Anzahl der untersuchten Nachbarknoten bei sechs bzw. 26.

Voraussetzung für die Durchführung des Prozesses ist ein Startpunkt innerhalb eines Körpers. Für das Finden eines Startpunktes ist der Einsatz des Verfahrens „*Punkt im Polyeder-Test*“ [162, 197] erforderlich. Einen effizienten Algorithmus für diesen Test stellt die von O'ROURKE vorgeschlagene Methode des *Ray Crossings* dar [162]. Hierbei wird, ausgehend von dem zu überprüfenden Knoten, ein Strahl in eine beliebige Richtung gesendet und die Anzahl der Schnitte zwischen Strahl und Polyeder gezählt. Ist die Anzahl ungerade, liegt der Kontrollpunkt innerhalb des Polyeders. Bei einem geraden Wert liegt der Knoten außerhalb des Körpers. Zusätzliche Regeln für diesen Algorithmus sind notwendig um geometrische Sonderfälle (kollineare Kante, Strahl trifft in Ecke) zu berücksichtigen.

Das generierte Voxelmodell wird zusammen mit den Randbedingungen an den Simulationskern zur numerischen Verarbeitung übermittelt. Die Übertragung erfolgt über das *MPI Protokoll* [151].

³⁶ Die als „Fluid“ deklarierten Voxel des Berechnungsgebietes werden über den numerischen Kern in den Simulationsprozess übernommen. Dabei kann der Kern nicht unterscheiden, ob der Fluidknoten innerhalb oder außerhalb eines Körpers liegt. Somit würden auch nicht sinnvolle Berechnungen von Luftströmungen innerhalb eines Objektes durchgeführt.

6.8 Schritt V: Das Berechnungsmodell

Bei dem Simulationskern zur Berechnung der Strömungs-, Feuchte- und Temperaturfelder handelt es sich um eine adaptierte Eigenentwicklung von NACHTWEY, TÖLKE und KRAFCZYK [156]. Luftströmungen werden auf einem uniformen *Finite-Differenzen-Gitter* über die *Lattice-Boltzmann-Methode* (siehe Abschnitt 6.8.1) berechnet. Der Rechenkern nutzt einen *D3Q19-Stern* (3 Raumdimensionen, 19 Verteilungsrichtungen), eine Modifikation [201] des *Multi-Relaxation-Time (MRT)* Ansatzes [54] und wurde mit *MPI-1* [151] parallelisiert.

Die Temperaturgleichung wird unter Einsatz eines Boltzmann-Ansatzes für *Advektions-Diffusionsgleichungen* mit einem *D3Q7 Modell* diskretisiert [83]. Die räumliche und zeitliche Auflösung gleicht hierbei der Auflösung der Strömungsberechnung. Über die Struktur erfolgt eine rein diffusive Berechnung auf einem klassischen Finite-Differenzen Gitter.

Die Simulation für die Feuchte erfolgt auf Basis des *zweiten Fickschen Gesetzes* mit einem Finite-Differenzen-Ansatz und nutzt eine reduzierte Raumdiskretisierung der Struktur analog der Temperaturberechnung. Die Zeitdiskretisierung entspricht der Strömungs- und Temperaturrechnung.

Das uniforme Voxel-Modell wird über den CFD-Kern rekursiv *bipartitioniert* [87]. Eine Partitionierung des Rechengebietes erfolgt zur Realisierung einer *parallelen Berechnung* [174] auf den Knoten des PC-Clusters (siehe Appendix A).

Der Simulationskern besteht aus mehreren Teilprozessen, welche mittels *MPI* [151] untereinander kommunizieren:

- Prozess zur Kommunikation über *Sockets* [208] mit dem Modellierer
- *Master-/Kollektorprozess* zur Kommunikation mit dem Visualisierer
- *Compute-/Slaveprozesse* (auf jedem Prozessor bzw. Knoten des PC-Clusters)

Die Socketkommunikation wurde unter Berücksichtigung der Kompatibilität in der Sprache *ANSI-C* [103] implementiert. Sie dient dem Empfang von Daten aus dem Modellierer. Zur Beschleunigung der Gesamtgeschwindigkeit sind die weiteren Programmteile des Rechenkerns in der maschinennahen Programmiersprache *Fortran* [109] implementiert.

6.8.1 Simulationen mit dem LB-Verfahren

Leistungsfähige Verfahren zur Simulation von thermischen Strömungen mit der *Lattice-Boltzmann-Methode* basieren unter anderem auf Arbeiten von [75, 130]. Diese bauen auf der *Boltzmann-Gleichung* auf, welche die Dynamik von *Aufenthaltswahrscheinlichkeiten von Teilchen* beschreibt. In [79] wurde basierend auf den Lattice-Gas Gleichungen gezeigt, dass hierzu eine Reduktion der Bewegungsfreiheitsgrade auf wenige Gitterrichtungen ausreicht, um die *Navier-Stokes Gleichung* mit einer Konvergenzrate zweiter Ordnung zu approximieren. Für die Anwendung bei schwach kompressiblen Fluiden und kleinen Machzahlen stellt die Lattice-Boltzmann-Methode einen sehr effizienten [82], gut skalierenden und leicht zu parallelisierenden Löser dar.

Da es sich hierbei um ein Verfahren zur direkten numerischen Simulation handelt, können nur relativ geringe, unrealistische *Reynolds-* und *Rayleighzahlen* erreicht werden. Eine aufwändige, aber genaue Methode zur Simulation turbulenter Strömungen ist die *Large-Eddy-Simulation* [125], welche auch mit Erweiterungen für thermische Strömungen eingesetzt werden kann.

An festen Wänden treten sehr dünne Strömungs- und thermische Grenzschichten auf, welche durch *Wandgesetze* abgebildet werden. Im LB-Kontext wurden solche Wandgesetze für thermische Grenzschichten adaptiert und validiert.

Das Lattice-Boltzmann-Verfahren ist noch sehr neu. Dies beweist der niedrige Marktanteil entsprechender Strömungslöser. *Power Flow* der Firma EXA [68] ist aktuell der einzige kommerzielle Code mit einem LB-Löser.

6.9 Schritt VI: Die Visualisierung

Ein Visualisierer im Bereich der Computergrafik dient der computergestützten, grafischen Repräsentation von Daten. Dieser Bereich beschreibt ein Gebiet, welches Methoden und Erkenntnisse der Informatik, Statistik, Data-Mining und Kognitionswissenschaft verwendet. Er dient der Verbesserung der *Mensch-Computer-Interaktion* [44]. In diesem Zusammenhang kann die bekannte Tabellenkalkulation MICROSOFT *Excel* [148] als Visualisierungs-Software verstanden werden. Rechnungen und Statistiken können mit dieser Software über Diagramme in anschaulicher Form *visualisiert* werden.

Bekannte Vertreter aus dem Bereich der *wissenschaftlichen Visualisierung* sind neben dem hier eingesetzten *AVS/EXPRESS* (*Advanced Visualization System*) [6] auch *Amira* [144] oder *COVISE* [214]. Zur Darstellung großer Datenmengen können Rechnerstrukturen für eine *parallele Visualisierung* in einem Verbund zusammen geschaltet werden [185]. Beispiele zur Darstellung von Daten aus numerischen Simulationen zeigen die Abbildungen in Kapitel 7.

AVS/EXPRESS dient der effizienten Darstellung großer Datenmengen. Das Programm besitzt eine mächtige Programmierschnittstelle zur Entwicklung eigener Visualisierungsmodule. Grundlage bildet hierbei die Programmiersprache C++ [108]. Die umfangreiche Modulbibliothek bietet *Isoflächen*, *Isovolumen* und *Stromlinien* zur anschaulichen Visualisierung von Luftströmungen. Module zur Darstellung der thermischen Behaglichkeit (PMV-Index, siehe Abschnitt 4.2) können eingebunden werden.

Die im Rahmen dieser Arbeit realisierte *interaktive Visualisierung* setzt ein automatisiertes, kontinuierliches Laden von Datensätzen der laufenden numerischen Berechnung voraus. Die sequenzielle Darstellung von Raumluftrömungen vermittelt dem Benutzer den Eindruck einen Film zu betrachten. Durch sein Einwirken über den Modellierer und den daraus resultierenden Einfluss auf die Simulation erhält das System seine *Interaktivität*.

Die Erweiterung des Visualisierungswerkzeugs *AVS/EXPRESS* ist zur Realisierung dieser Funktionen erforderlich. Folgende Softwaremodule wurden implementiert, um die Notwendigkeit eines manuellen Benutzereingriffes zu verhindern:

Aufgrund der großen Mengen an numerischen Ergebnisdaten sind bei der Kommunikation zwischen Simulationskern und Visualisierer hohe Bandbreiten sowie effiziente Kommunikationsprotokolle notwendig. Das vom FORSCHUNGSZENTRUM JÜLICH entwickelte Programm VISIT (*Visualization Interface Toolkit*; [65, 66]) erfüllt diese Anforderungen. Es basiert auf der TCP Socket-Kommunikation und wurde für den Transfer von Daten numerischer Simulationen in einer Computational Steering Umgebung entwickelt. Die Abbildung 6.20 beschreibt die Kopplung von Simulationskern und dem Visualisierer AVS/EXPRESS über das Programm VISIT.

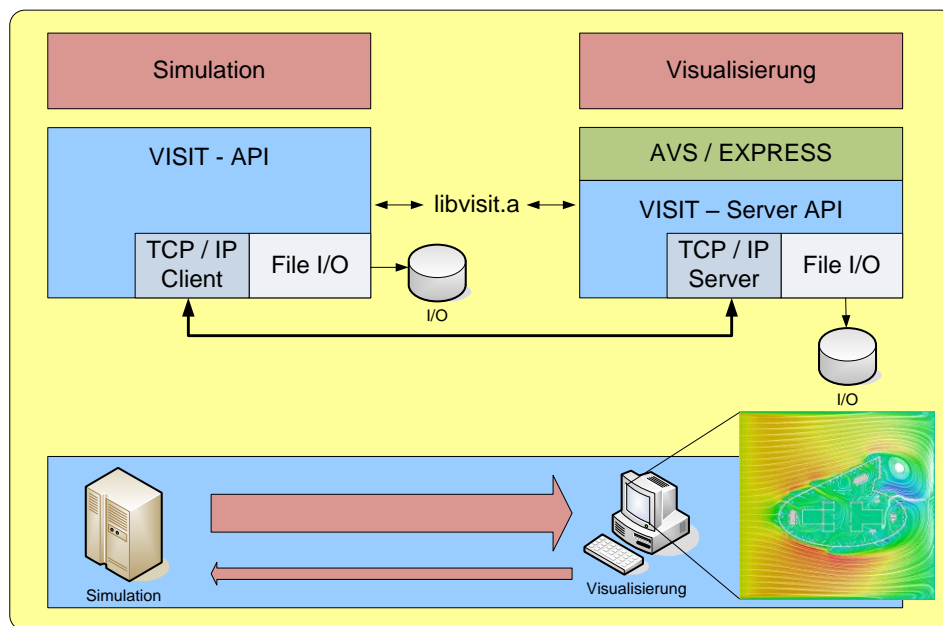


Abbildung 6.20: Funktionsweise des Kopplungs-Moduls VISIT

Die Ergebnisdaten der Simulation werden nach einer Anzahl von Rechenschritten als vierdimensionales Feld automatisiert an das Visualisierungsmodul gesendet. Das Feld beinhaltet die Raumkoordinaten sowie einen Vektor der Strömungsdaten. Der Visualisierer bezieht keine geometrischen Informationen aus dem von VISIT übertragenen Vektorfeld. Dies erfolgt über eine vom Modellierer erzeugte Datei im *UCD* (*Unstructured Cell Data*; [5]) Dateiformat (siehe Abbildung 6.9). Hierzu ist ein zusätzliches AVS-Modul erforderlich, welches simultan die Datei importiert und als Oberflächenmodell visualisiert. Die Objektgeometrie wird dabei über skalierungsfähige Knotenpunktdaten übertragen und ist somit auflösungsunabhängig. Dieses ermöglicht eine präzise Darstellung der geometrischen Objekte innerhalb des Simulationsgebietes. Simulationsdaten und geometrisches Modell werden durch den Visualisierer zusammengeführt und über eine *Virtual Reality-Anlage* ausgegeben (siehe Appendix A).

6.10 Schlussfolgerung und Ausblick

Automatisierte Modellableitung für numerische Simulationen. Der in diesem Kapitel vorgestellte Ansatz zeigt Möglichkeiten zur Steuerung einer numerischen Simulation über einen automatisierten Prozess. Durch die interaktive Kopplung bekannter Softwareprodukte und den Wegfall des Pre-Processings zwischen konstruktiver Planung und numerischer Analyse kann die Akzeptanz eines praktisch orientierten Bauingenieurs stark erhöht werden. Verstärkt wird dies durch die Nutzung einer im Bauwesen bekannten und akzeptierten CAD-Applikation als Modellierungswerkzeug. Diese Voraussetzungen bergen den Erfolg für einen weitreichenden Einsatz numerischer Simulationen im Bauwesen. Hierfür ist Software mit einfachem Aufbau sowie intuitiver Benutzerführung von enormer Bedeutung. Diese Forderung wird bei einem kooperativen Einsatz verschärft.

Aktuelle kommerzielle Softwareapplikationen für die numerische Simulation gelten als Insellösungen, da Pre-Processing, Simulation und Visualisierung in manuellen Teilschritten erfolgen muss. Für eine Automatisierung dieser Teilschritte bieten die Programme keine ausreichend standardisierten Schnittstellen. Ebenfalls sind die Einarbeitungszeiten für einen praxisorientierten Fachplaner aufgrund der nicht-intuitiven Benutzeroberflächen, der Komplexität der Programme und der Anforderung an vorausgesetztes Fachwissen zu hoch.

Gebietszerlegung des Rechengebietes. Der aktuell eingesetzte Algorithmus der Gebietszerlegung für eine parallele Berechnung arbeitet mit einer rekursiven Bisektionierung, welche das Rechengebiet in Teile gleicher Größe unterteilt. Hierbei wird eine Knotentypisierung (Fluid, Struktur, Randbedingung) und eine sich daraus ergebende ungleiche Rechenlast nicht berücksichtigt [90]. Als sinnvoll erscheint die Umstellung auf einen Gebietszerleger zur Berücksichtigung von Knotentypen. Dies führt zur Erzeugung von Gebieten gleichen Rechenaufwandes aber ungleicher Größe. Lösungen bieten hier die Algorithmen von GUN [88, 91] (für würfelförmige Gebiete) oder der *graphenbasierte Zerleger METIS* [115] (für Gebiete in Form einer beliebigen ausgearteten Berandung).

Simulationskern. Der eingesetzte CFD-Kern setzt die Struktur eines uniformen Gitters für das Berechnungsmodell voraus. Ein logischer Schritt wäre die Implementierung eines adaptiven Modells [53], welches sich im Einsatz mit dem *Lattice-Boltzmann Verfahren* als sehr effizient herausgestellt hat [82].

Oktalbaum, intelligentes Lokalisierungsverfahren. Der hier vorgestellte Ansatz zur Komplexitätsreduzierung für die Lokalisierung von geometrischen Informationen in einer Baumstruktur stellt eine sinnvolle Verbesserung herkömmlicher Oktalbaum-Algorithmen dar. Doch dieser Ansatz setzt weiterhin einen vollständigen Neuaufbau des Rechengebietes bei lokalen Geometrieänderungen voraus. Eine höhere Effizienz bieten adaptive Oktalbäume. Hierbei können über *Schnitte* innerhalb des Oktalbaums Unterbäume erzeugt werden. Der oberste Knoten des Unterbaumes wird hierbei zum Root-Oktanten. Dieses Verfahren ermöglicht bei lokalen geometrischen Änderungen des Simulationsgebietes die Diskretisierung von Teilbereichen eines Oktalbaums ohne einen Neuaufbau des gesamten Oktalbaums zu fordern (siehe [78] sowie die Arbeiten von MUNDANI [153, 154]).

Kapitel 7

Simulationsbeispiele

7.1 Klimatechnische Auslegung eines Gebäudes

Bei der technischen Gebäudeausrüstung kann die optimale Auslegung einer Klimaanlageinstallation über die Messung der thermischen Behaglichkeit im Raum objektiv bewertet werden. In der Planungsphase wird die Frage nach einer optimalen Auslegung und Anordnung von Klimatelementen nur über die numerische Simulation beantwortet. Die Behaglichkeit wird über den in Abschnitt 4.2 beschriebenen PMV-Index definiert und kann als skalarer Wert der Simulation direkt visualisiert werden.

In dem hier dargestellten Szenario werden vier variierende Ausbaustufen eines Klimaanlageentwurfs für einen Büroraum auf ihre Effizienz überprüft. Diese ergibt sich aus den Parametern der thermischen Behaglichkeit über den PMV-Index, den Installationskosten der Anlage und der benötigten Zeit der Anlage zur Herstellung eines behaglichen Klimas im Raum.

Die erste und zweite Ausbaustufe der Anlage besitzen Klimaanlageeinlässe an den Fensterflächen auf der Längs- oder Querseite des Raumes. In der dritten Ausbaustufe sind Einlässe über die gesamte Fensterfront des Gebäudes verteilt. Die vierte Ausbaustufe weist gegenüber der dritten Version zusätzliche Quellluftauslässe in einer rasterförmigen Anordnung auf dem Boden auf. Alle Ausbaustufen weisen Klimaanlageauslässe an den Gebäudekernen auf. Die Abbildung 7.1 stellt den Aufbau der vier Ausbaustufen der Klimaanlageinstallation über den Grundriss des Stockwerks dar. Als geometrisches Modell dient ein Stockwerk des UNIQA-Towers in Österreich [211].

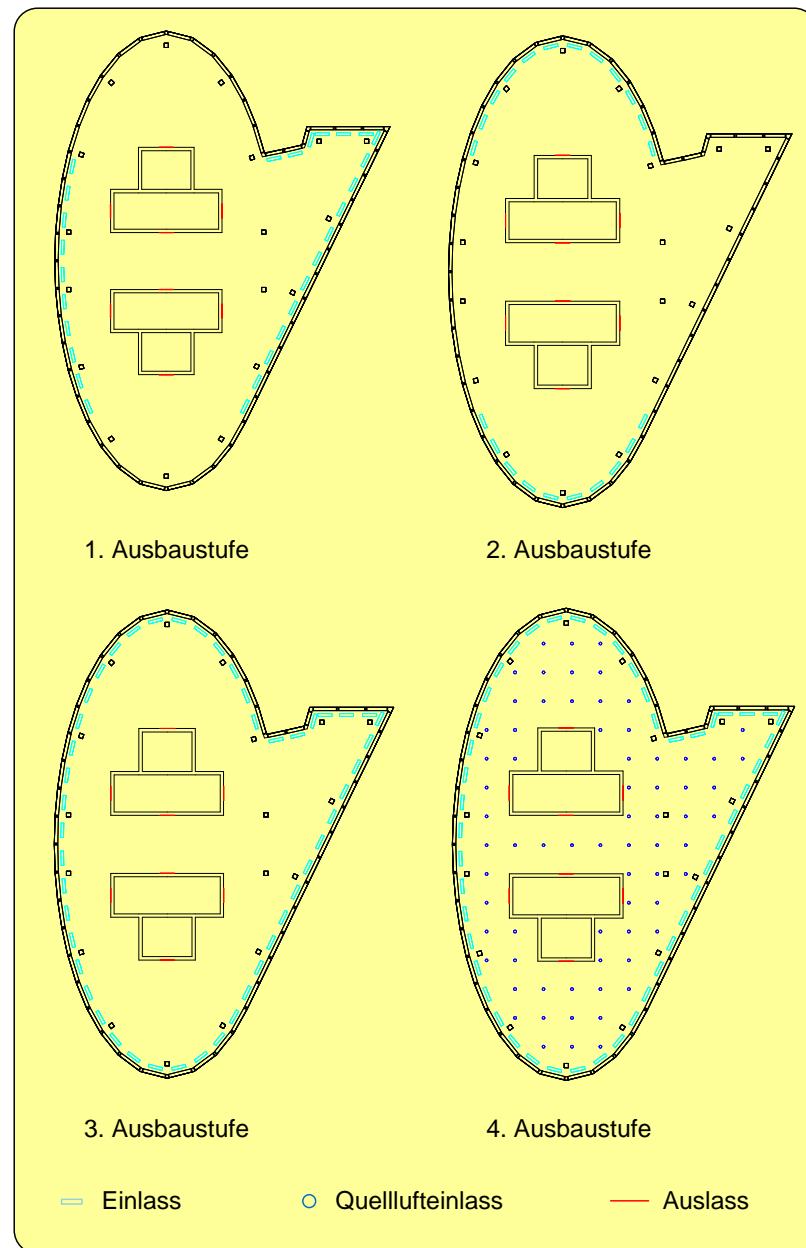


Abbildung 7.1: Verschiedene Ausbaustufen einer HVAC-Installation für ein Gebäude.

Für die Simulation wird die initiale Raumtemperatur auf 10°C gesetzt und soll über die Klimaanlageinstallation auf einen Wert von 22°C angehoben werden. Die Luftgeschwindigkeit der Klimaanlageeinlässe liegt bei 3 m/s , die Geschwindigkeit der Quelllufteinlässe bei $0,5\text{ m/s}$. Die Abbildung 7.2 zeigt auf der rechten Seite einen Schnitt durch das Temperaturfeld. Die Farbskala definiert kalte Bereiche mit der Farbe Blau und warme Bereiche mit der Farbe Rot. Die gewünschte Temperatur von 22°C wird über die grüne Farbe dargestellt. Auf der linken Seite markiert die grüne Isofläche Zonen des Raums mit akzeptablem Klimakomfort. Das Raumklima, welches sich nach 60 Minuten einstellt, wird wiedergegeben.

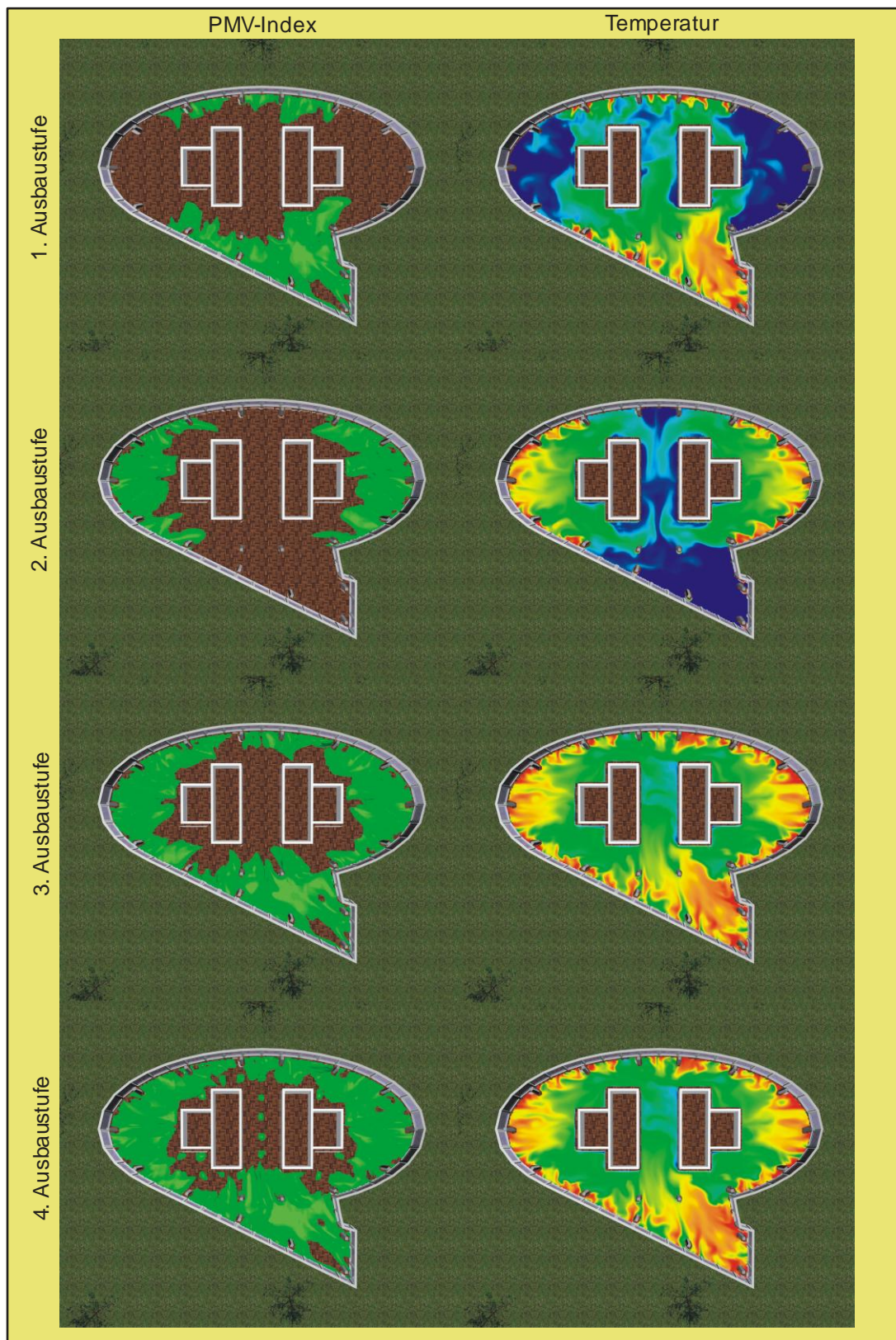


Abbildung 7.2: Visualisierung von PMV-Index und Temperaturfeld verschiedener Ausbaustufen einer Klimaanlageinstallation

Die Simulation zeigt deutliche Defizite bei der ersten und zweiten Ausbaustufe. Große Bereiche des Stockwerks können über diese Anlagen nicht erwärmt werden. Die Temperatúrausbreitung der dritten und vierten Ausbaustufe ist nahezu identisch. Auch die thermische Behaglichkeit im Raum variiert zwischen den beiden Versionen nur minimal. Unter Berücksichtigung der Installationskosten, welche in der vierten Ausbaustufe deutlich höher ausfallen, kann die dritte Version als optimale Anlagenauslegung bestimmt werden.

Die numerische Simulation erlaubt Einblicke in die zeitliche Entwicklung der berechneten Vorgänge. Die direkte Visualisierung erlaubt eine detaillierte Analyse der Anlageninstallationen zur Detektion von Schwachstellen einzelner Komponenten oder einer ungünstigen Positionierung von Klimaanlagelementen. Die Abbildung 7.3 veranschaulicht über eine Sequenz die zeitliche Entwicklung des Raumklimas für die vierte Ausbaustufe.

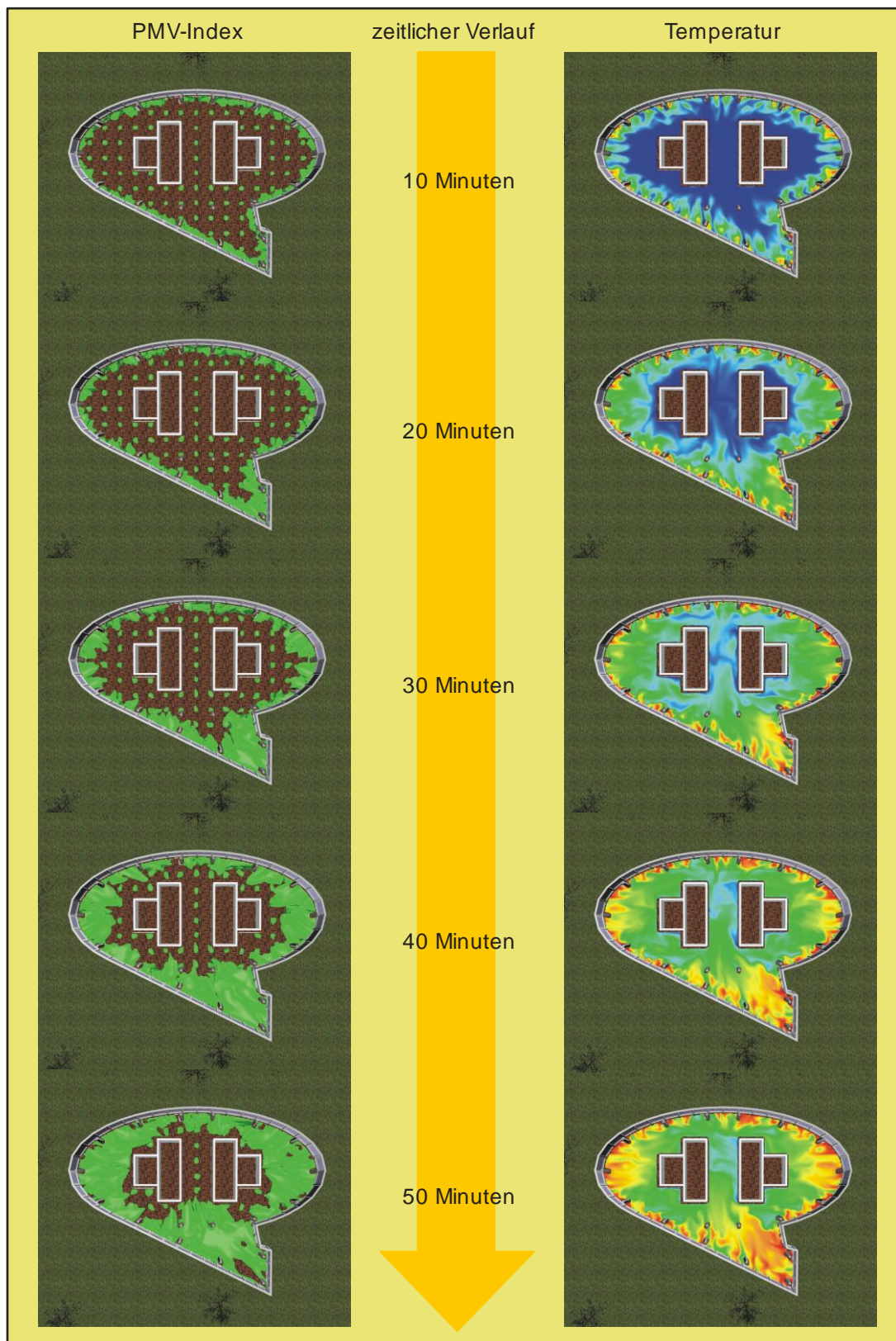


Abbildung 7.3: Zeitliche Entwicklung von Temperaturfeld und PMV-Index für die 4. Ausbaustufe

7.2 Klimaanlagesteuerung über MSR-Technik

Das hier dargestellte Szenario beschreibt die Simulation einer Klimaanlageinstallation mit agentengesteuerten Regelwerken. Als Simulationsgebiet wurde der in Abbildung 7.4 dargestellte Büroraum gewählt.

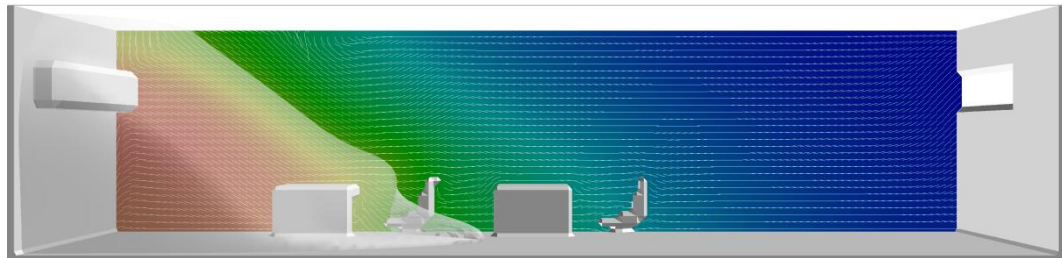


Abbildung 7.4: Büroraum mit geregelter Klimaanlageauslass über HVAC-Agenten

Der Raum besitzt einen Klimaanlageeinlass über eine Wandinstallation auf der linken Seite und einen Klimaanlageauslass auf der rechten Seite. Zur permanenten Erfassung der Raumtemperatur ist ein Messfühler auf dem linken der beiden Schreibtische positioniert. Der Klimaanlageeinlass sowie der Messfühler werden über HVAC-Agenten emuliert.

Die Steuerung des Luftstroms erfolgt über die Temperatur, die Geschwindigkeit und den Einfallswinkel. Für die Regelung wurden integrale Regler ohne Totzeit verwendet. Die Temperaturwerte der einströmenden Luft sind auf minimal 18 °C und maximal 30 °C begrenzt. Die maximale Luftgeschwindigkeit ist unter Berücksichtigung der Behaglichkeit im Raum auf 0,35 m/s limitiert. Der Einfallswinkel der injizierten Luft wird über eine Lamellensteuerung emuliert. Der Winkel ist hierbei abhängig von der Temperaturdifferenz zwischen Raumtemperatur und der Temperatur der einströmenden Luft. Bei einem Temperaturunterschied von $\Delta T \geq 6$ °C wird der Luftstrom in einem Winkel von 5° in Richtung des Fussbodens eingeblasen. Bei sehr geringer Temperaturdifferenz erfolgt eine Ablenkung des Luftstroms mit einem Winkel von 175° in Richtung der Decke des Raumes. Die kontinuierliche Anpassung des Einfallswinkels erfolgt mit einem linearen Verhalten in Teilschritten zwischen 5° und 175° [176].

Zur Validierung der simulierten Anlagenregelung sind die Mess- und Steuerwerte der Temperatur über die Laufzeit der Simulation dargestellt. Die in der Abbildung 7.5 dargestellte simulierte Regelung der Temperatur zeigt eine qualitative Übereinstimmung mit den in der Abbildung 7.6 dargestellten Ergebnissen einer Referenzanlage aus der Literatur [25] auf.

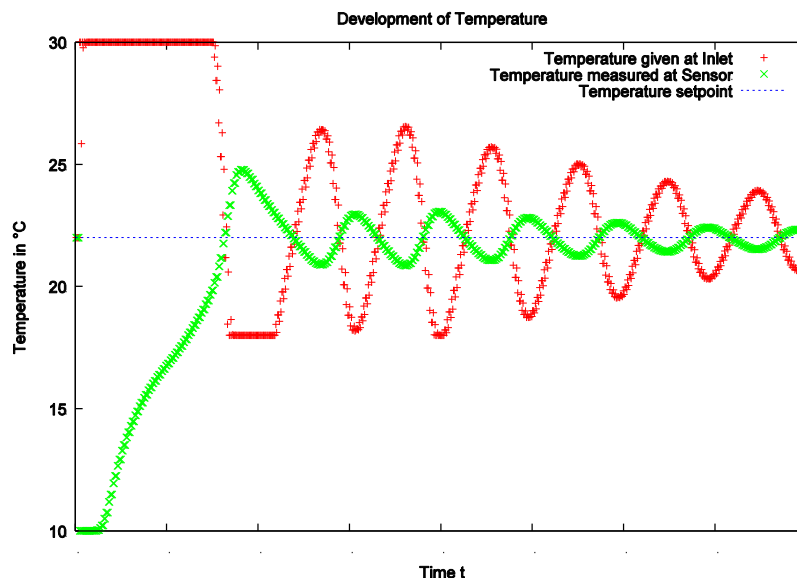


Abbildung 7.5: Verhalten eines I-Reglers: Ergebnisse der Simulation

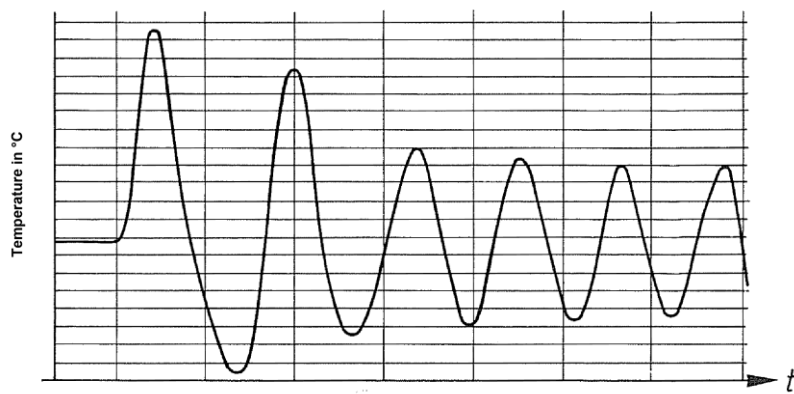


Abbildung 7.6: Verhalten eines I-Reglers: Ergebnisse aus der Literatur

Ein I-Regler summiert eine Regelabweichung über die Zeit auf. Je länger eine Regelabweichung an einem Regler ansteht, desto größer wird die Stellgröße. Liegt bei einem I-Regler der Istwert unterhalb des Sollwertes, wird der Stellgrad kontinuierlich aufgebaut. Es ist bei Erreichen des Sollwertes aufgrund von Verzögerungen der Strecke wahrscheinlich, dass der Stellgrad zu groß ist. Baut sich der Stellgrad zu schnell auf, ist das entstehende Stellsignal zu groß und ein zu hoher Istwert wird erreicht. Die Folge bei dem Einsatz eines I-Reglers ist daher oft eine Schwingung um den Sollwert. Dies ist der Fall, wenn die Integrierzeit zu klein gewählt wurde [184].

Zum Erreichen einer signifikanten Ausprägung des Regelkreises wurde die Integrierzeit des I-Reglers sehr klein gewählt. Beide Diagramme zeigen schwach gedämpfte Oszillationen, welche das erläuterte Verhalten eines I-Reglers charakterisieren.

7.3 Luftströmung innerhalb einer Flugzeugkabine

Die Simulation von Luftströmungen und Klimaanlageinstallationen kann neben dem Bauingenieurwesen auch in anderen Gebieten effizient eingesetzt werden. Der große Funktionsumfang des eingesetzten Modellierers erlaubt Simulationen für beliebig komplexe geometrische Modelle. Ein interessantes Anwendungsgebiet ist die Simulation von Luftströmungen in Flugzeugkabinen über climatechnische Installationen. Simulationen dieser Art sind sehr aufwändig und werden häufig an Realmodellen durchgeführt [34, 86].

Bei dem hier vorgestellten Simulationsbeispiel soll die qualitative Ausprägung der Luftströmungen aus den Ergebnissen des Software-Prototyps mit Ergebnissen aus der Literatur verglichen werden. Entsprechend einer Veröffentlichung des Unternehmens BOEING beschreibt die Abbildung 7.7 die typische Luftbewegung innerhalb einer Kabine eines Passagierflugzeugs [92].

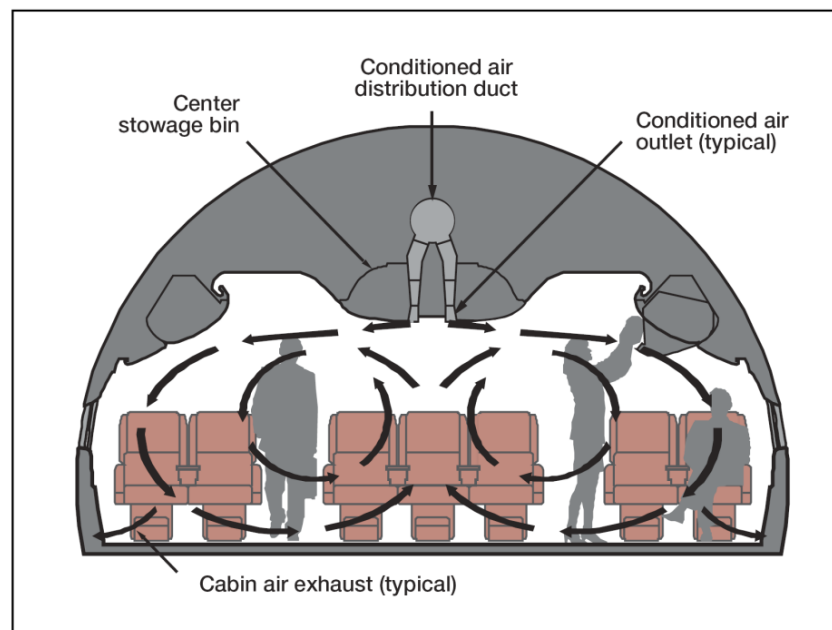


Abbildung 7.7: Typische Luftbewegung innerhalb einer Flugzeugkabine (Quelle: BOEING)

Ein dreidimensionales Modell einer Flugzeugkabine sowie der Klimaanlageinstallation wurde über den Modellierer des Software-Prototyps nachgebildet und die in der Abbildung 7.8 dargestellte Luftausbreitung simuliert.

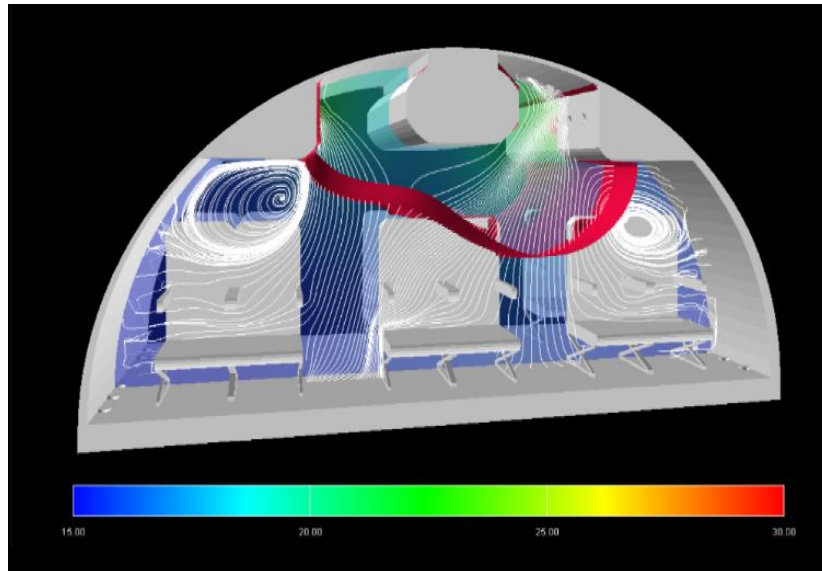


Abbildung 7.8: Simulation der Luftbewegungen innerhalb einer Flugzeugkabine über den Software-Prototyp

Die in der Farbe Weiß dargestellten Stromlinien zeigen einen Verlauf mit einer signifikanten Wirbelbildung an den Innenseiten der Flugzeugkabine. Dieser entspricht dem der Abbildung 7.7 aus der Literatur. Die rot dargestellte Isofläche beschreibt die Temperaturentwicklung bei Erwärmung der Kabine über die Klimaanlageeinlässe.

Kapitel 8

Schlussfolgerung und Ausblick

Die vorliegende Arbeit beschreibt Techniken, die zur Realisierung eines Software-Prototyps für eine interaktive Entwurfs- und Simulationsumgebung eingesetzt wurden. In den vorangegangenen Kapiteln wurden die Produktmodellanbindung, die interaktive Modellierung, die HVAC-Simulation, der Einsatz immersiver Technologien und der automatisierte Modelltransfer von einem Produktmodell in ein Datenmodell für die numerische Simulation erörtert. Diese Kapitel schließen mit einer themenbezogenen Diskussion ab und werden in Kapitel 8 in einer allgemeinen Schlussfolgerung und einem Ausblick zusammengefasst.

8.1 Schlussfolgerung

Planungsprozesse im Bauwesen. Der hier vorgestellte Software-Prototyp kann das benötigte Zeitaufkommen bei der Durchführung von Optimierungsprozessen, welche in der Praxis an manuelle und iterative Interaktionen während der Planungsphase gebunden sind, wesentlich reduzieren. Die effektive Kommunikation zwischen den Fachplanern und die direkte Visualisierung der Ergebnisdaten reduziert Entwurfs- und Konstruktionsfehler nachweislich. Kooperationsmodelle zeigen erfolgreich Methoden auf prototypischer Basis zur Realisierung eines domänenübergreifenden Austauschs von Daten auf Basis eines IFC Produktmodells [70, 120]. Durch Nutzung bekannter kommerzieller Applikationen als Mensch-Maschine-Schnittstelle kann die Akzeptanz für diese neuen Technologien reduziert werden. Die Einführung eines automatisierten Pre-Processings gewährleistet die notwendige Benutzerfreundlichkeit bei der Einführung neuer Software-Systeme.

Der im Rahmen dieser Arbeit eingesetzte Computational Steering Ansatz bietet ideale Voraussetzungen für kooperative Planungs- und Optimierungsprozesse und bildet eine Schnittstelle zwischen Fachdomänen. Die Kosten für das im Rahmen dieser Arbeit eingesetzte Computational Steering System liegen bei mehreren hunderttausend Euro. Für mittelständische und kleine Bauunternehmen ist somit die Anschaffung von Hardware für die benötigte Rechenleistung zum aktuellen Zeitpunkt nicht wirtschaftlich. Alternativen bieten hier die Mietlösungen, bei denen PC-Cluster „on demand“ für ein aktuell anstehendes Bauvorhaben kurzzeitig genutzt werden können [142]. Das dieser Trend Zukunft hat, beweisen hohe Investitionen namhafter Firmen wie beispielsweise die der Firma AMAZON, welche ihren Supercomputing-Service "*Elastic Compute Cloud*" (EC2) [8] auf Stundenbasis weltweit über das Internet mit hohen Bandbreiten anbietet [137].

Stellung der numerischen Gebäudesimulation im Bauwesen. Die numerische Simulation wird zum heutigen Zeitpunkt im Bereich des Bauwesens nur wenig eingesetzt. Bei standardisierten Bauvorhaben mit normaler Raumgeometrie und Dimensionierung werden CFD Simulationen in der Regel nicht durchgeführt. Dies ist unter anderem auch durch die nicht nach HOIA geregelte Finanzierung der Simulationskosten zu begründen. Vereinzelt finden hier vergleichsweise numerisch einfache Simulationen auf Basis von Zonenmodellen (beispielsweise für die Energiebilanzierung) Anwendung.

Ein Paradigmenwechsel ist noch längst nicht vollzogen, Teilaspekte bauspezifischer Problemstellungen werden jedoch verstärkt über numerische Ansätze gelöst. Bei speziellen Großprojekten zeigt die Simulation von Raumluftrömungen hohes Potenzial. Die vereinfachten Berechnungsmethoden der HOIA [188], welche bei der Kühllastberechnung standardmäßig nach VDI 2078 [213] durchgeführt werden, neigen bei komplexer Geometrie und hohen Raumhöhen zur Überdimensionierung des benötigten Luftvolumenstroms und somit der gesamten HVAC Anlageninstallation. Bei großen Sälen mit zusätzlicher Lichttechnik im Deckenbereich reichen diese Berechnungsmethoden für eine effiziente Anlagenauslegung nicht aus, numerische Simulationen sind hier notwendig [172]. Nach heutigen Erfahrungen der Industrie können die dabei entstehenden Kosten für die Simulation bei diesen Bauvorhaben im Durchschnitt durch nachfolgende Kosteneinsparungen bei Installation und Betrieb um den Faktor fünf bis zehn kompensiert werden.

Das IFC Produktmodell als Datenbasis. Das IFC Produktmodell dient als Datenbasis für den im Rahmen dieser Arbeit entwickelten Software-Prototyp. Für den Prototyp erforderliche Metainformationen von Bauteilen konnten über dieses Modell in einem nicht standardisierten Format abgebildet werden. Die IFC bieten in ihrem aktuellen Umfang nicht die notwendigen Spezifikationen für die standardisierte Integration erforderlicher Objektattribute zur numerischen Simulation und zur HVAC-Technik. Die über die IFC zur Verfügung gestellte HVAC Domain deckt nur Teilbereiche der für das HVAC-Agenten Framework erforderlichen Bauteilspezifikationen ab. Eine CFD Domain existiert innerhalb des IFC Produktmodells nicht. Nachteile der IFC gegenüber dem AEC Produktmodell finden sich bei der Abbildung geometrischer Objekte. Es existieren Defizite bei der Darstellung gekrümmter Oberflächen.

Zur Einführung des IFC Produktmodells als Datenbasis für das Bauwesen müssen neben einem fehlenden Innovationsengagement innerhalb dieses Sektors noch grundlegende Probleme gelöst werden. Diese beziehen sich unter anderem auf das Problem der Datensicherheit, hervorgerufen durch ein allgemeines Misstrauensverhältnis zwischen den Planungsbeteiligten. Durch fehlende Zertifizierungs- und Sicherheitsmechanismen sind Daten nicht gerichtsverwertbar. Auch trägt die Komplexität eines Produktmodells mit seiner hohen Informationsdichte zur Erhöhung der Akzeptanz bei. Relevante Informationen können erst über das Filtern des Modells dargestellt werden.

Simulation klimatechnischer Anlagen. Die Integration von Mess- Steuer- und Regeltechniken erweitert die Möglichkeiten der HVAC Simulation und erhöht die Genauigkeit der erzielten Ergebnisse. Das transiente Verhalten von Regelkreisen und Regelgliedern kann intuitiv analysiert und optimiert werden. In Kombination mit dem Computational Steering Ansatz kann der Hausleittechniker Parameter der HVAC-Anlagensteuerung bei direkter Visualisierung der Auswirkungen variieren.

In der heutigen Praxis können Anlagen nur mit einem sehr hohen Aufwand getestet und optimiert werden. Zur Bestimmung der Strömungsverläufe in einem Raum oder Gebäude werden in der Regel Tracer-Gase [102] eingesetzt. Eine Optimierung der Regeltechnik erfolgt im Allgemeinen nach der vollständigen Installation des Klimaanlageansystems und setzt bereits die Fertigstellung des Gebäudes voraus.

Durch Nutzung einer Simulationsumgebung stehen dem Fachplaner nun Möglichkeiten zur Verfügung, Installationskonfigurationen zu testen, welche sich stark von Standardsystemen unterscheiden und teilweise zu neuen und innovativen Konstruktionslösungen führen können [195]. Auf einfache Weise sind Vergleiche komplexer Alternativsysteme mit Standardkonfigurationen und Standardanordnungen unter Berücksichtigung von Energieverbrauch und Raumbehaglichkeit möglich.

Die Simulation klimatechnischer Anlagen beschränkt sich nicht auf das Gebiet des Bauingenieurwesens. Durch die Mächtigkeit des eingesetzten Modellierers sind auch Szenarien mit komplexen geometrischen Modellen wie beispielsweise Flugzeugkabinen möglich (siehe Abschnitt 7.3). Praxisorientierte Simulationen in diesem Bereich sind heutzutage immer noch sehr aufwändig und werden häufig an Realmodellen durchgeführt [34, 86].

Software-Agenten. Für die Realisierung einer HVAC Simulation innerhalb einer interaktiven Computerumgebung würde die Agenten-Technologie eingesetzt. Über diesen Ansatz eröffnet sich eine intuitiv verständliche Sicht auf Software und Vorgänge. Dies reduziert die Akzeptanz der Fachplaner des Bauingenieurwesens mit dem Umgang neuer, innovativer Techniken. Hierbei kann das zu Grunde gelegte Softwaresystem der Agententechnologie als Organisationsstruktur angesehen werden, bei dem einzelne Softwareeinheiten (Agenten) unter Einhaltung bestimmter Vorschriften (Regeln und Normen) mit einem definierten Verhalten autonom agieren um Konflikte zu lösen. Diese Einheiten können dynamisch in übergeordnete Organisationseinheiten übergehen und als Team bestimmte Aufgaben und Verpflichtungen übernehmen (beispielsweise Messung und Regelung).

8.2 Ausblick

Über den Software Prototyp können zum jetzigen Stadium Probleme der Strömungsmechanik und Regelungstechnik gelöst werden. Zur Berücksichtigung von Anforderungen weiterer Fachplaner können *zusätzliche Simulationskerne* aus dem Bereich der Statik, Elektrotechnik, Akustik oder Lichtsimulation mit dem virtuellen Entwurfsraum gekoppelt werden. Projekte wie *DIVERCITY*³⁷ [10, 180] zeigen hier entsprechende Ansätze.

Die Kopplung verschiedener Simulationsdisziplinen zeigt einen sinnvollen Schritt zur integralen Betrachtung und Bewertung von Gebäuden. Dies unterstreicht die im Jahr 2007 eingeführte DIN 18599 „Energetische Bewertung von Gebäuden – Berechnung des Nutz-, End- und Primärenergiebedarfs für Heizung, Kühlung, Lüftung, Trinkwarmwasser und Beleuchtung“ [62]. Das Nachweisverfahren erfordert infolge seiner Komplexität einen hohen Aufwand durch Datensammlung und Berechnung. Da es sich bei dem DIN-Nachweis um ein statisches Verfahren handelt, bildet dieser die Realität unzureichend ab und liefert somit recht ungenaue Ergebnisse. Der Einsatz von numerischen, gekoppelten Simulationswerkzeugen ist hier sinnvoll [129].

Die im Rahmen dieser Arbeit vorgestellte Software-Applikation stellt prototypische Methoden zur Durchführung von Entwurfs- und Optimierungsprozessen zur Verfügung. Für einen praktischen Einsatz des virtuellen Entwurfsraums sind Erweiterungen im Bereich des numerischen Kerns sowie des HVAC-Agenten Frameworks notwendig. Die Erweiterungskonzepte wurden in den vorangegangenen Kapiteln genauer erläutert. Verwiesen sei auf zukünftige Arbeiten von NACHTWEY und BINDICK, welche die Entwicklung des Software-Prototyps im Rahmen eines praxisorientierten DFG-Projektes unter dem Fokus der Klimatisierungstechnik fortführen.

³⁷ DIVERCITY, Distributed Virtual Workspace for enhancing Communication within the Construction Industry

Appendix A

Hardware

Computational Steering Systeme stellen durch die Einbindung von komplexer Simulation, Datenanalyse, Visualisierung und Nachbearbeitung hohe Anforderungen an die eingebundene Hardware. Zur Erfüllung dieser Forderungen wurde im Rahmen dieses Projektes eine *Virtual Reality Umgebung* zur Wiedergabe der Visualisierung und Modellierung und ein *PC-Cluster* für die CFD-Simulation eingebunden.

Die VR-Umgebung arbeitet nach dem Prinzip der Stereoprojektion. Auf eine transparente Leinwand werden in Rückprojektionstechnik zwei überlagerte Bilder mittels Videoprojektoren für das linke und rechte Auge projiziert. Die Trennung der Bilder erfolgt mittels gerichteter Polfilter und entsprechender Filterbrillen. Der an die VR-Anlage angeschlossene Workstation-PC (Visualisierungsrechner) berechnet beide Bilder separat auf einer High-End-Grafikkarte mit zwei separaten Ausgängen, an denen die Projektoren angeschlossen sind. Die Steuerung der Visualisierungssoftware erfolgt über einen dreidimensionalen *Motion Tracker* [12], welcher über ein magnetisches Feld sechs translatorische Freiheitsgrade überwacht. Der Tracker ist über einen seriellen Anschluss mit dem Visualisierungsrechner verbunden.

Der PC-Cluster setzt sich aus einem Doppelprozessor-Master-Knoten und 59 Doppelprozessor Compute-Knoten zusammen, welche über ein Hochleistungs-Netzwerk [155] verbunden sind. Der PC-Cluster ist im mehrere Kilometer entfernten Maschinensaal des Rechenzentrums der TU Braunschweig aufgestellt und ist zum jetzigen Zeitpunkt der schnellste Rechner im Braunschweiger Raum.

Zur Gewährleistung einer effizienten Kommunikation wurde die Verbindung zwischen Workstation-PC und PC-Cluster über eine Glasfaserleitung mit einer Bandbreite von einem Gigabit gewählt.

Der Workstation-PC ist zusätzlich über eine 100 Megabit-Verbindung an das Internet angebunden, um über die in Abschnitt 5.3 beschriebene Multi-User-Umgebung weitere Fachplaner in den Optimierungsprozess einzubinden. Der Einsatz zusätzlicher Hardware zur Gewährleistung von Audio- und Video-Konferenzen erhöht die Effizienz des Planungsprozesses. Der Aufbau des Hardwareverbundes wird über die Abbildung 8.1 dargestellt. Die nachfolgenden Abbildungen zeigen Bilder der eingesetzten Hardware.

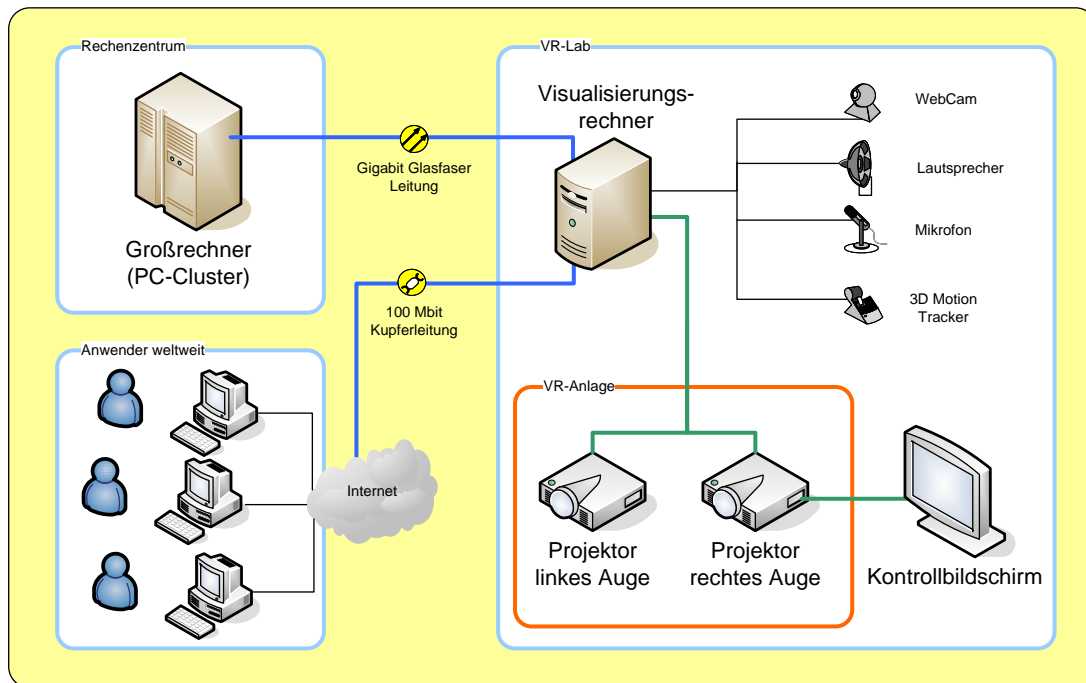


Abbildung 8.1: Hardware und Infrastruktur des Software Prototyps

Anzahl CPUs:	122 Prozessoren, AMD Opteron, 64 Bit, davon: Master-Knoten: 2 Stk. x 2.0 GHz Compute-Nodes: 94 Stk. x 1.4 GHz plus 24 Stk. x 2.0 GHz Login-Node: 2 Stk. x 2.0 GHz
gesamter Hauptspeicher:	221 GByte, davon: Master-Knoten: 16 GB Compute-Nodes: 41 Stk. je 3 GB plus 14 Stk. je 4 GB plus 4 Stk. je 6 GB Login-Node: 2 GB
Festplattenspeicher:	4,665 TeraByte, (TB) davon: 1,75 TB per NFS 2,915 TB lokal auf den Compute-Nodes
Peakperformance	367 GFLOPS
Bandbreite Netzwerk	250 MByte/sek. (Myrinet2000)

Tabelle 8.1: Technische Daten des PC-Clusters

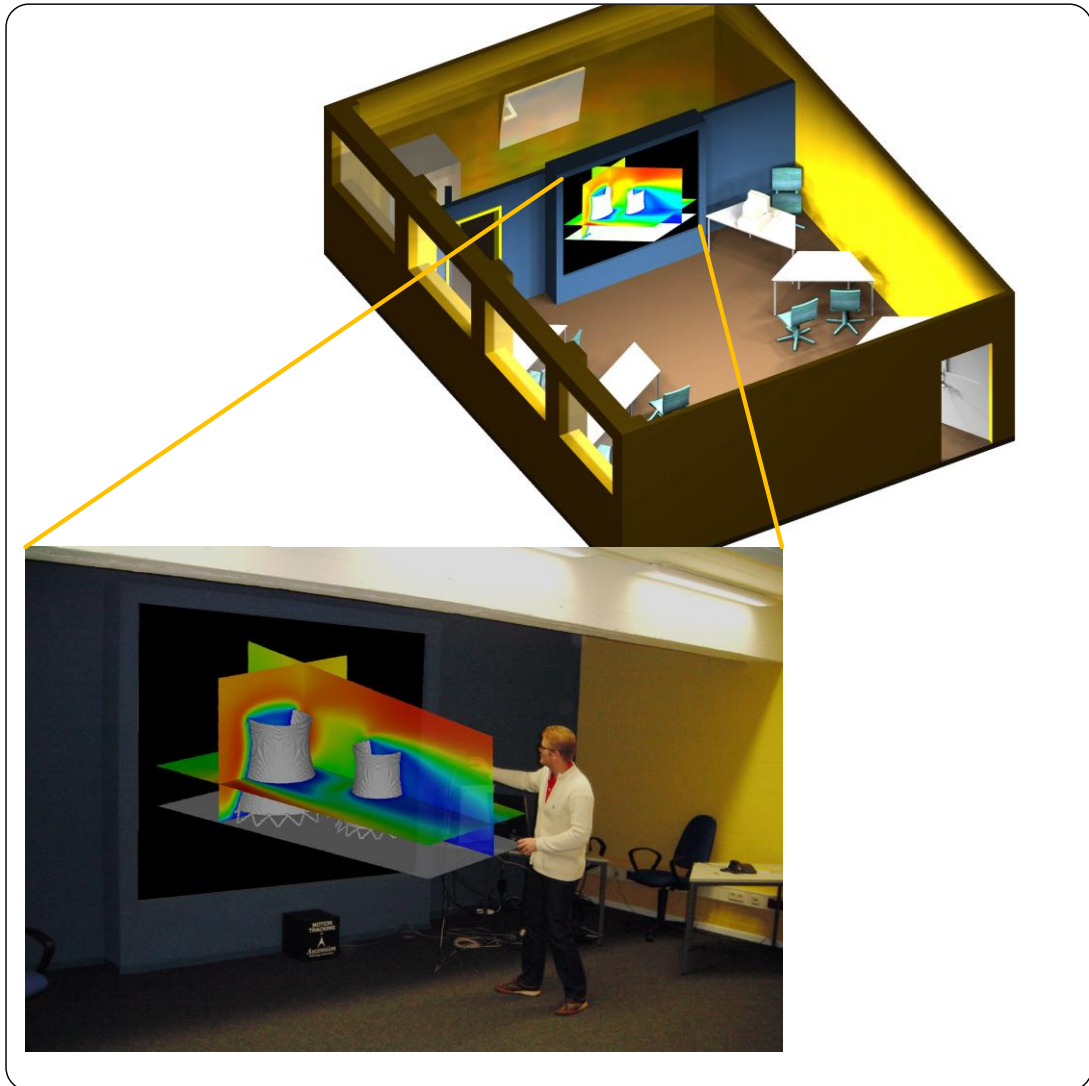


Abbildung 8.2: Virtual Reality Anlage des Instituts



Abbildung 8.3: PC-Cluster im Rechenzentrum der TU Braunschweig

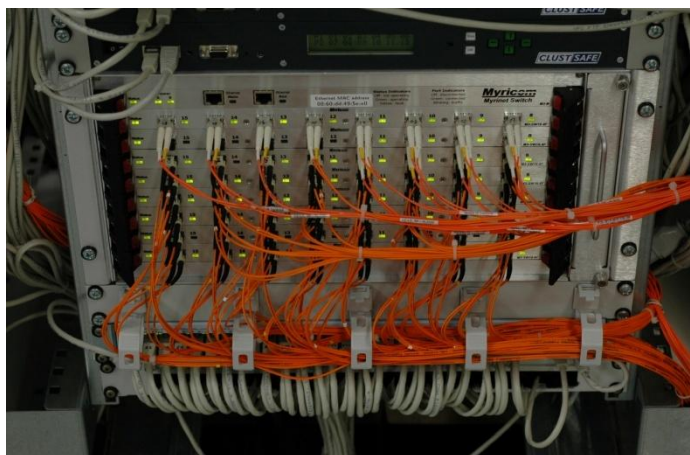


Abbildung 8.4: Ein Hochgeschwindigkeitsnetzwerk (Myrinet) verbindet die Knoten des PC-Clusters

Literatur

- [1] Adobe (2006). *AEC solution brief - Automate and accelerate your business process for increased competitive advantage*.
http://www.adobe.com/products/acrobat/pdfs/AEC_SB_FINAL.pdf
- [2] Adobe (2006). *Solutions for architecture, engineering, and construction*.
<http://www.adobe.com/products/acrobat/solutions/aec/index.html>
- [3] Adobe (2007). *Adobe Acrobat*, Version: 8.0.
<http://www.adobe.com/de/products/acrobat/>
- [4] Adobe (2007). *Adobe Acrobat 3D - Publish, share, review, and mark up 3D designs in Intelligent Documents*, Version: 8.0.
<http://www.adobe.com/products/acrobat3d/>
- [5] Advanced Visual Systems Inc. (AVS) (1991). *Unstructured Cell Data (UCD)*, Typ: Dateiformat.
<http://www.iavsc.org>
- [6] Advanced Visual Systems Inc. (AVS) (2005). *AVS/EXPRESS*, Version: 7.0, Typ: Visualisierungs-Software.
<http://www.avs.com>
- [7] Akenine-Möller, T. (2001). *Fast 3D triangle-box overlap testing*. In *Journal of Graphics Tools*. Vol: 6(1): S. 29-33.
- [8] Amazon Web Services (2006). *Amazon Elastic Compute Cloud (Amazon EC2) - Limited Beta*.
<http://www.amazon.com/gp/browse.html?node=201590011>
- [9] Anderl, R. und Trippner, D. (2000). *STEP - Standard for the Exchange of Product Model Data*. Stuttgart, Germany: B. G. Teubner Verlag.
- [10] Arayici, Y., Ahmed, V. und Aouad, G. (2006). *A Requirement Engineering Framework For Integrated Systems Development For The Construction Industry*. In *ITcon*. Vol: 11: S. 35-55.
- [11] Arlt, M. (2000). *Agentenbasierte Systemarchitekturen für Produktdaten-Management-Systeme*. Aachen, Germany: Shaker-Verlag.
- [12] Ascension Technology Corp. (2007). *Flock of Birds*.
<http://www.ascension-tech.com/products/flockofbirds.php>

-
- [13] ASHRAE - American Society of Heating Refrigerating and Air Conditioning Engineers (2006). *ASHRAE Handbook for HVAC and R technology, Standard 62.1, 90.1, 34, 55*.
<http://www.ashrae.org/>
- [14] ATLAS Projekt (1992).
<http://www.newcastle.research.ec.org/esp-syn/text/7280.html>
- [15] Autodesk (2003). *OMF Developer's Guide 3.3*, Version: 3.3.
<http://adn.autodesk.com>
- [16] Autodesk (2005). *Autodesk Architectural Desktop*, Version: 2005, Typ: AEC/CAD.
<http://www.autodesk.com>
- [17] Autodesk (2005). *Autodesk Architectural Desktop - User's Guide*.
- [18] Autodesk (2005). *Autodesk AutoCAD*, Version: 2005, Typ: CAD.
<http://www.autodesk.com>
- [19] Autodesk (2005). *Autodesk ObjectARX Developer's Guide*, Version: 2005.
<http://adn.autodesk.com>
- [20] Autodesk (2005). *ObjectARX/DBX/OMF Application Wizard for AutoCAD 2005*, Version: 2005, Typ: Entwickler-Werkzeug.
<http://adn.autodesk.com>
- [21] Autodesk (2006). *Autodesk Building Systems*, Version: 2006, Typ: building design and construction documentation application.
<http://www.autodesk.com/buildingsystems>
- [22] Autodesk (2006). *Revit Building*, Version: 9.
<http://www.autodesk.de/adsk/servlet/index?siteID=403786&id=6644243>
- [23] Autodesk (2006). *White Paper - Das Gebäudeinformationsmodell in der Praxis*.
<http://www.autodesknews.de/inewsn/docs/informationsmodell.pdf>
- [24] Autodesk (2007). *Autodesk AutoCAD Architecture*, Version: 2008, Typ: CAD.
<http://www.autodesk.de/adsk/servlet/index?siteID=403786&id=8958683>
- [25] Bach, H. et al. (1983). *Regelungstechnik in der Versorgungstechnik*. Karlsruhe, Germany: Verlag C.F. Müller GmbH.
- [26] Bellifemine, F., Caire, G., Rimassa, G., Poggi, A. und Trucco, T. (2007). *JADE (Java Agent DEvelopment Framework)*.
<http://jade.tilab.com/>

- [27] Berghen, F.V. (2006). *Small, simple, cross-platform, free and fast C++ XML Parser*, Version: 2.12.
<http://www.applied-mathematics.net/>
- [28] Bernstein, P. (2006). *Gebäude wie Antos*. In AUTOCAD Magazin. Vol: 4/06.
http://buildingsmart.de/pdf/ps_2006-10.pdf
- [29] Bindick, S. (2006). *Ein Prototyp zur rechnergestützten Kraneinsatzplanung mit dem Autodesk Architectural Desktop 2005*. Studienarbeit, CAB - Institut für Computeranwendungen im Bauingenieurwesen, TU Braunschweig, Germany.
- [30] Blaschke, C., Dayal, M. und Liebich, T. (2006). *Integrierte 3D-Planung bei AUDI - Schnittstelle mit Potenzial*. In industrieBAU. Vol: 3.
http://buildingsmart.de/pdf/ps_2006-09.pdf
- [31] Blaschke, C. und Liebich, T. (2006). *Vortrag LAI-Industrietag - IFC-Projekterfahrung bei Audi*.
http://buildingsmart.de/pdf/9-IAI_Industrietag_Audi.pdf
- [32] Böhms, M. und Storer, G. (1994). *ATLAS - Architecture, methodology and Tools for computerintegrated Large Scale engineering*. In JSPE-IFIP WG 5.3 Workshop, DIISM'93. Tokyo, Japan.
- [33] Borrmann, A., Wenisch, P., Treeck, C.v. und Rank, E. (2006). *Collaborative computational steering: Principles and application in HVAC layout*. In Integrated Computer-Aided Engineering. Vol: 13: S. 1-16.
- [34] Bosbach, J., Kühn, M., Rütten, M. und Wagner, C. (2006). *Mixed Convection in a Full Scale Aircraft Cabin Mock-Up*. In ICAS 2006 - 25th International Congress of the Aeronautical Sciences. Hamburg, Germany.
<http://elib.dlr.de/43927>
- [35] Bresenham, J.E. (1965). *Algorithm for Computer Control of a Digital Plotter*. In IBM Systems Journal. Vol: 4(1): S. 25-30.
- [36] Bröker, H. (2001). *Integration von geometrischer Modellierung und Berechnung nach der p-Version der FEM*. Dissertation, Lehrstuhl für Bauinformatik, TU München, Germany.
- [37] Brüggemann und Holz (2004). *Quadtree based hydroinformatics simulation systems: A component-oriented finite volume toolkit for shallow water equations*. In 6th International Conference on Hydroinformatics. Singapore: World Scientific Publishing Co. Pte. Ltd.
- [38] Bundesministerium für Raumordnung Bauwesen und Städtebau (1996). *Dritter Bericht über Schäden an Gebäuden*.

- [39] Bungartz, H.-J., Griebel, M. und Zenger, C. (1996). *Einführung in die Computergraphik*. Braunschweig/Wiesbaden, Germany: Friedr. Vieweg&Sohn Verlagsgesellschaft mbH.
- [40] Burdea, G. und Coiffet, P. (2003). *Virtual Reality technology*. 2. Ed: John Wesley & Sons.
- [41] Cakmakci, O. und Rolland, J. (2006). *Head-worn displays: a review*. In Journal of Display Technology. Vol: 2(3): S. 199-216.
- [42] Camps, T., Kühling, S. und Rosenberger, G. (2006). *Einführung in die mengentheoretische und die algebraische Topologie*, Ed. Heldermann.
- [43] Center for the Study of Complex Systems at the University of Michigan (2007). *SWARM - Agent- and Individual-based Modeling Resources*.
http://www.swarm.org/wiki/ABM_Resources
- [44] Chen, C. (2006). *Information Visualization. Beyond the Horizon*. 2. Ed. Berlin, Germany: Springer.
- [45] Cohen, J., Lin, M.C., Manocha, D. und Poamgi, M. (1996). *I-COLLIDE: An interactive and Exact Collisions Detection System for Large Scale Environment*. In Symposium on Interactive 3D Graphics.
- [46] Collis, J., Ndumu, D. und Buskirk, C.v. (2007). *The Zeus Agent Building Toolkit*.
<http://labs.bt.com/projects/agents/zeus/>
- [47] COMBI Projekt (1995).
<http://www.cib.bau.tu-dresden.de/combi/>
- [48] Comer, D. (2000). *Internetworking with TCP/IP. Principles, Protocols, and Architectures*. Prentice Hall.
- [49] Coutinho, M.G. (2001). *Dynamic Simulations of Multibody Systems*. Computers/Desktop Publishing: Springer.
- [50] Crouse, B. (2003). *Lattice-Boltzmann Strömungssimulation auf Baumdatenstrukturen*. Dissertation, Lehrstuhl für Bauinformatik, TU München, Germany.
- [51] Crouse, B., Krafczyk, M., Kühner, S., Rank, E. und Treeck, C.v. (2002). *Indoor air flow analysis based on lattice Boltzmann methods*. In Int. J. Energy and Buildings, Elsevier.
- [52] Crouse, B. und Kühner, S. (1999). *Simulation und Visualisierung von Windströmungen um Bauwerke*. In Forum Bauinformatik. Darmstadt, Germany: Fortschrittsberichte VDI, Düsseldorf, VDI Verlag.

- [53] Crouse, B., Rank, E., Krafczyk, M. und Tölke, J. (2002). *A LB-based approach for adaptive flow simulations*. In Int. J. of Modern Physics. Vol: B(17): S. 109-112.
- [54] D’Humieres, D., Ginzburg, I., Krafczyk, M., Lallemand, P. und Luo, L.-S. (2002). *Multiple-relaxation-time lattice Boltzmann models in three dimensions*. In Phil. Trans. R. Soc. Lond. A. Vol: 360: S. 437-451.
- [55] digital IMAGE - Virtual Reality Systems (2007). *Cube 3D² active stereo projector*.
<http://cube3d2.de/>
- [56] DIN Deutsches Institut für Normung e.V. (1981). *DIN 19225, Messen, Steuern, Regeln - Benennung und Einteilung von Reglern*. Beuth Verlag GmbH.
- [57] DIN Deutsches Institut für Normung e.V. (1994). *DIN 19226, Teil 1 - Leittechnik; Regelungstechnik und Steuertechnik; Allgemeine Begriffe*. Beuth Verlag GmbH.
- [58] DIN Deutsches Institut für Normung e.V. (1994). *DIN 19226, Teil 2 - Leittechnik; Regelungstechnik und Steuertechnik; Begriffe zum Verhalten dynamischer Systeme*. Beuth Verlag GmbH.
- [59] DIN Deutsches Institut für Normung e.V. (1994). *DIN 19226, Teil 3 - Leittechnik; Regelungstechnik und Steuertechnik; Begriffe zum Verhalten von Schaltsystemen*. Beuth Verlag GmbH.
- [60] DIN Deutsches Institut für Normung e.V. (1994). *DIN 19226, Teil 4 - Leittechnik; Regelungstechnik und Steuertechnik; Begriffe für Regelungs- und Steuerungssysteme*. Beuth Verlag GmbH.
- [61] DIN Deutsches Institut für Normung e.V. (1994). *DIN 19226, Teil 5 - Leittechnik; Regelungstechnik und Steuertechnik; Funktionelle Begriffe*. Beuth Verlag GmbH.
- [62] DIN Deutsches Institut für Normung e.V. (2005). *DIN 18599, Energetische Bewertung von Gebäuden - Berechnung des Nutz-, End- und Primärenergiebedarfs für Heizung, Kühlung, Lüftung, Trinkwarmwasser und Beleuchtung*. Beuth Verlag GmbH.
- [63] Eastman, C.M. (1999). *Building Product Models*. CRC Press LLC.
- [64] EDS Corp. (2006). *EDS Parasolid*.
<http://www.eds.com>
- [65] Eickermann, T. et al. (2005). *Steering UNICORE applications with VISIT*. In Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. Vol: 363(Nr. 1833 / August 15, 2005): S. 1855-1865.
- [66] Eickermann, T., Frings, W. und Visser, A. (2001). *VISIT - a Visualization Interface Toolkit*, Version: 2.0beta, Typ: Kommunikation.
<http://www.fz-juelich.de/zam/visit/>

- [67] Ekholm, A. (2005). *ISO 12006-2 and IFC – Prerequisites For Coordination Of Standards For Classification And Interoperability*. In ITcon. Vol: 10: S. 275-289.
<http://www.itcon.org/2005/19>
- [68] Exa GmbH (2006). *PowerFLOW Technology Overview*.
<http://www.exa.com/newsite/PowerFLOW/powerflowtechnology.html>
- [69] Fahrig, T., Krafczyk, M., Nachtwey, B. und Tölke, J. (June 2006). *Enhanced Computational HVAC Simulations using Software Agent Technology*. In Joint International Conference on Computing and Decision Making in Civil and Building Engineering. Montreal, Canada.
- [70] Fahrig, T., Lähr, A. und Niggel, A. (2005). *Multidisziplinäre Analyse und kooperative Simulation auf Grundlage eines Bauteilsmodells*. In 17. Forum Bauinformatik Cottbus, Germany.
- [71] Fahrig, T. und Nachtwey, B. (2004). *Ein Prototyp für verteilte, interaktiv-kooperative Simulationen zur Beschleunigung von Entwurfszyklen im konstruktiven Ingenieurbau*. In Forum Bauinformatik. Braunschweig, Deutschland.
- [72] Fahrig, T., Tölke, J. und Krafczyk, M. (2005). *Agent-based Measuring, Control and Regulation Techniques for HVAC-Simulations*. In Building Simulations Conference. Montreal, Canada.
- [73] Fakespace Systems Inc. (2007). *Hi-PowerWall™*.
<http://www.fakespace.com/hiPowerWall.htm>
- [74] Farin, G. (1997). *Curves and surfaces for computer aided geometric design*. 4. Ed: Academic Press.
- [75] Filippova, O. und Hänel, D. (1998). *Lattice-BGK Model for Low Mach Number Combustion*. In Int. Journal of Modern Physics C. Vol: 9, N 8: S. 1439.
- [76] Foley, J.D. (1995). *Computer Graphics: Principles and Practice in C*: Addison-Wesley Professional.
- [77] Folino, G. und Spezzano, G. (1999). *Bioremediation Experiments using Interactive Computational Steering on High Performance Computers*. In Fifth European SGI/Cray MPP Workshop. Università della Calabria, Rende, Italy.
<http://www.cineca.it/mpp-workshop/fullpapers/spezzano/spezzano.htm>
- [78] Frank, A.C. (2000). *Organisationsprinzipien zur Integration von geometrischer Modellierung, numerischer Simulation und Visualisierung*. Dissertation, Lehrstuhl für Informatik, TU München.

- [79] Frisch, U. et al. (1986). *Lattice gas hydrodynamics in two and three dimensions*. In Complex Systems. USA.
- [80] G.E.M. Team Solutions und Inopso GmbH (2005). *IFC-Utility 2x for ADT 2005*. <http://www.inopso.com/>
- [81] Gamma, E., Helm, R. und Johnson, R.E. (2004). *Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley.
- [82] Geller, S., Krafczyk, M., Tölke, J., Turek, S. und Hron, J. (2006). *Benchmark computations based on Lattice-Boltzmann, Finite Element and Finite Volume Methods for laminar Flows*. In Computers and Fluids. Vol: 35: S. 888-897.
- [83] Ginzburg, I. (2005). *Equilibrium-type and Link-type Lattice Boltzmann models for generic advection and anisotropic-dispersion equation*. In Advances in Water Resour. Vol: 28: S. 1171-1195.
- [84] Gottschalk, S. (2000). *Collisions detection using Oriented Bounding Boxes*. PhD Thesis, Chapel Hill, University of Carolina, USA.
- [85] Grosso, R., Wechsler, K., Ertl, T. und Schäfer, M. (1996). *Computational steering and visualization for multiblock multigrid flow simulations*. In High-performance computing and networking: International Conference and Exhibition. Brussels, Belgium: H. Liddell (u.a.).
- [86] Günther, G. et al. (2006). *Experimental and numerical simulation of idealized aircraft cabin flows*. In Aerospace Science and Technology. (10): S. 565 - 573. <http://elib.dlr.de/19275>
- [87] Gun, A. (2002). *Ein Gebietszerleger für parallele Lattice Boltzmann Applikationen auf der Basis von rekursiv-bipartiven Algorithmen für equidistante teilgefüllte 3D-Gitter*. Diplomarbeit, Lehrstuhl für Bauinformatik, TU München, Germany.
- [88] Gun, A. (2002). *Entwicklung eines Gebietszerlegers für parallele Lattice Boltzmann Applikationen auf der Basis von rekursiv-bipartiven Algorithmen für equidistante teilgefüllte 3D-Gitter*. Diplomarbeit, Lehrstuhl für Bauinformatik, TU München, Germany.
- [89] Hennig, A. (1997). *Die andere Wirklichkeit*. Addison-Wesley.
- [90] Hinkelmann, R. (1997). *Parallelisierung eines Lagrange-Euler-Verfahrens für Strömungs- und Stofftransportprozesse in Oberflächengewässern*. Dissertation, Inst. für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Germany.
- [91] Hossain, F. (2006). *Implementation of a Communication-Scheme for an Explicit Finite-Difference Algorithm of Higher Order for Block-Structured Domain Decompositions*.

- Studienarbeit, CAB- Institut für Computeranwendungen im Bauingenieurwesen, TU Braunschweig, Germany.
- [92] Hunt, E.H. und Space, D.R. (1994). *The Airplane Cabin Environment - Issues Pertaining to Flight Attendant Comfort*. In International In-flight Service Management Organization Conference. Montreal, Canada.
- [93] IAI's Software Implementer Support Group (2006). *IFC2x3-Software Implementation Certification*.
<http://129.187.85.207:8080/fb02/IAI-ISG-Portal/>
- [94] IAI - International Alliance for Interoperability (2000). *IFC 2x / ifcXML*.
<http://www.iai.org.uk>
- [95] IAI - International Alliance for Interoperability (2006).
<http://www.iai-international.org/>
- [96] IAI - International Alliance for Interoperability (2006). *Data Modelling Using EXPRESS-G for IFC Development*.
http://www.iai-international.org/Model/documentation/Data_Modelling_Using_EXPRESS-G_for_IFC_Development.pdf
- [97] IAI - International Alliance for Interoperability (2006). *Industry Foundation Classes Documentation - IFC2x Edition 3*.
http://www.iai-international.org/Model/R2x3_final/index.htm
- [98] IEEE Foundation for Intelligent Physical Agents (2007). *FIPA-OS Agent Toolkit*.
<http://sourceforge.net/projects/fipa-os/>
- [99] Industrieallianz für Interoperabilität e.V. (2006).
<http://buildingsmart.de/>
- [100] Information Sciences Institute - University of Southern California (1981). *RFC: 793 - TRANSMISSION CONTROL PROTOCOL*.
<http://tools.ietf.org/html/rfc793>
- [101] Innova AirTech Instruments (2002). *Thermal Comfort*. Ballerup, Denmark.
- [102] Innova AirTech Instruments (2002). *Ventilation Measurements and other Tracer-Gas Applications*. Ballerup, Denmark.
- [103] ISO - International Organization for Standardization (1990). *ANSI C, Version: 90*.
<http://www.ansi.org>

- [104] ISO - International Organization for Standardization (1994). *ISO 10303-1 (STEP): Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles*.
- [105] ISO - International Organization for Standardization (2002). *ISO 10303-21 (STEP): Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure*.
- [106] ISO - International Organization for Standardization (2003). *Deutsche Fassung prEN ISO Standard 7730 "Analytische Bestimmung und Interpretation der thermischen Behaglichkeit durch Berechnung des PMV- und des PPD-Indexes und der lokalen thermischen Behaglichkeit"*.
- [107] ISO - International Organization for Standardization (2004). *ISO 10303-11 (STEP): Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual*.
- [108] ISO - International Organization for Standardization JTC1/SC22/WG21 (1998). *ISO/IEC 14882: The programming language C++*.
- [109] ISO - International Organization for Standardization WG5 (1996). *ISO/IEC DIS 1539-1 (Fortran 95)*, Version: 95.
- [110] ISO (International Standards Organization) (1995). *EN ISO Standard 9920 "Ergonomics of the thermal environment - Estimation of the thermal insulation and evaporative resistance of a clothing ensemble"*.
- [111] ISO (International Standards Organization) (1998). *EN ISO Standard 7726 "Ergonomics of the thermal environment - Instruments for measuring physical quantities"*.
- [112] ISO (International Standards Organization) (2002). *prEN ISO Standard 8996 "Ergonomics of the thermal environment - Determination of metabolic heat production"*.
- [113] Jakob, R. und Weiß, G. (2004). *Agentenorientierte Softwareentwicklung. Methoden und Tools*. Berlin: Springer.
http://www.amazon.de/gp/reader/3540000623/ref=sib_dp_pt/303-5284920-1547451#reader-page
- [114] Jaksch, S. (2001). *Facettierung dreidimensionaler Gebiete und Gittergenerierung unter Verwendung von Octree-Datenstrukturen*. Diplomarbeit, Lehrstuhl für Bauinformatik, TU München, Germany.
- [115] Karypis, G. und Kumar, V. (1998). *A fast and high quality multilevel scheme for partitioning irregular graphs (METIS)*. In *SIAM Journal on Scientific Computing*. Vol: 20(1): S. 359-392.

- [116] Katranuschkov, P. und Scherer, R.J. (1996). *Schema Mapping And Object Matching: A Step-Based Approach To Engineering Data Management In Open Integration Environments*. In CIB-W78 Workshop „Construction on the Information Highway“, Bled’96.
- [117] Kohl, J.A., Wilde, T. und Bernholdt, D.E. (2006). *Cumulus: Interacting with High-Performance Scientific Simulations, for Visualization, Steering and Fault Tolerance*. In International Journal of High Performance Computing Applications. Vol: 20: S. 255-285.
- [118] Krafczyk, M. (2001). *Gitter-Boltzmann-Verfahren - von der Theorie zur Anwendung*. Habilitationsschrift, Lehrstuhl Bauinformatik, TU München, Germany.
- [119] Krafczyk, M. (2004). *Computersimulation und Technologiefortschritt – eine symbiotische Beziehung*. In Journal der Deutschen Akademien der Wissenschaften.
- [120] Krafczyk, M. et al. (2006). *A generalized product-model based framework for multidisciplinary sensitivity analysis and optimization in civil engineering*. In Joint International Conference on Computing and Decision Making in Civil and Building Engineering. Montreal, Canada.
- [121] Krafczyk, M., Lehmann, P., Schulz, M., Tölke, J. und Rank, E. (2000). *Direct Computation of Multi-Phase Flow in Reconstructed Soil*. In CIMENICS. Puerto La Cruz, Venezuela.
- [122] Krafczyk, M. und Rank, E. (1994). *High-Performance Computing and Networking: Supercomputing on a Workstation Cluster: A Parallelized Lattice-Gas Solver for Transient Navier-Stokes-Flow*. Lecture Notes in Computer Science, Ed. H. Gentzsch: Springer
- [123] Krafczyk, M. und Rank, E. (1995). *A Parallelized Lattice-Gas Solver for Transient Navier-Stokes-Flow: Implementation and Simulation Results*. In Int. Journal for Num. Methods in Eng. Vol: 38 S. 1243-1258.
- [124] Krafczyk, M., Schulz, M., Rank, E. und Cerrolaza, M. (1998). *Simulation and Visualization of transient 3D-Flow in arteries with an artificial heart valve using Lattice-Boltzmann Methods*. In Journ. of Biomech. Vol: 31: S. 453-462.
- [125] Krafczyk, M., Tölke, J. und Luo, L.S. (2002). *LES simulations based on a multiple relaxation time LB model*. In Int. J. of Modern Physics. Vol: B.
- [126] Kramer, B. (2000). *ObjectARX Primer*. Thomson Learning, Autodesk's Programmer Series.
- [127] Kühner, S. und Krafczyk, M. (2000). *Virtual Fluids - An environment for integral visualisation and analysis of CAD and simulation data*. In 5th International Fall Workshop Vision, Modeling, and Visualization. Saarbrücken, Germany.
- [128] Kuhn, G. (1999). *Stereofotografie und Raumbildprojektion*. vfv Verlag.

- [129] Lahme, A. (2006). *Integrale Analyse von Gebäuden durch Simulation*. In BauSIM 2006. München, Germany.
- [130] Lallemand, P. und Luo, L.-S. (2000). *Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability*. In Physical Review Vol: E 61: S. 6546-6562.
- [131] Lavedrine, I. und Woolf, D. (2004). *CFD Analysis in the Built Environment - Internal Ventilation in BENCHmark*.
<http://www.nafems.org>
- [132] Leetaru, K. (2007). *The CAVE at NCSA*.
<http://cave.ncsa.uiuc.edu/>
- [133] Leinonen, J., Kähkönen, K. und Retik, A. (2003). *New construction management practice based on the virtual reality technology*. In 4D CAD and Visualization in Construction: Developments and Applications: A.A. Balkema Publishers.
- [134] Lietsch, S., Zabel, H. und Berssenbrügge, J. (2007). *Computational Steering of Interactive and Distributed Virtual Reality Applications*. In ASME 2007 International Design Engineering Technical Conference & Computers and Information in Engineering Conference. Las Vegas, USA.
- [135] Linden, E., Hellstrom, J., Cehlin, M. und Sandberg, M. (2001). *Virtual reality presentation of temperature measurements on a diffuser for displacement ventilation*. In Proceedings of the 4th International Conference on Indoor Air Quality and Energy Conservation in Buildings. Changsha, Hunan, China.
- [136] Lipton, L. (1997). *StereoGraphics Developers' Handbook - Background on Creating Images for CrystalEyes and SimulEyes*: StereoGraphics Corporation.
<http://www.stereographics.com>
- [137] List, A. (2006). *Amazon bietet Supercomputing als Web-Service*, in *presstext Nachrichtenagentur GmbH*: Wien, Austria.
<http://www.presstext.de/pte.mc?pte=060825023>
- [138] Malkawi, A. und Primikiri, E. (2002). *Visualizing building performance in a multi-user virtual environment*. In Proceedings of the ACSA International Conference. Havana, Cuba.
- [139] Malkawi, A.M. (2003). *Immersive building simulation*. In Advanced Building Simulation. S. 217-246.
- [140] Maßmann, J. (2005). *Mobile Kommandozentrale - Server-Fernsteuerung mit Smartphone und VNC*. heise mobil - Heise Zeitschriften Verlag GmbH & Co. KG.
<http://www.heise.de/mobil/artikel/59931>

- [141] McAuley, C. (2000). *Programming AutoCAD 2000 using ObjectARX*. Thomson Learning, Autodesk's Programmer Series.
- [142] MEGWARE Computer GmbH (2006). *Rechenleistung nach Bedarf*.
<http://www.megware.de/pdf/rechenleistungnachbedarf.pdf>
- [143] Meissner, U.F. und Ruben, J. (2003). *Middleware-based modelling and simulation of geotechnical structures*. In Proceedings of the CIB W78's 20th International Conference of Information Technology For Construction. University of Auckland, New Zealand: AMOR, R.
- [144] Mercury Computer Systems (2007). *Amira - Advanced 3D Visualization and Volume Modeling*, Version: 4.1, Typ: Visualisierungs-Software.
<http://www.amiravis.com/>
- [145] Microsoft (2002). *Visual Studio .NET*, Typ: Entwickler-Werkzeug.
<http://msdn.microsoft.com/vstudio/>
- [146] Microsoft (2006). *COM/COM+ (Component Object Model)*.
<http://microsoft.com>
- [147] Microsoft (2006). *Microsoft Foundation Classes (MFC)*, Version: 7.0.
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclib/html/mfc_Class_Library_Reference_Introduction.asp
- [148] Microsoft (2007). *Microsoft Office Online*, Typ: Büroanwendung.
<http://office.microsoft.com/de-de/default.aspx>
- [149] Microsoft (2007). *Microsoft Windows Plattform*, Typ: Betriebssystem.
<http://www.microsoft.com/germany/windows/default.mspx>
- [150] Microsoft MSDN (2006). *ActiveX Controls Gallery*.
<http://activex.microsoft.com/activex/activex/>
- [151] MPI-Forum (1997). *MPI - Message Passing Interface*, Version: 1.2.
<http://www.mpi-forum.org>
- [152] MuM - Mensch und Maschine Software AG (2006). *RoCAD Lüftung und Heizung*.
<http://mum.de/mum/default.asp?subportal=Haustechnik>
- [153] Mundani, R.-P. (2006). *Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben*. Shaker Verlag.
- [154] Mundani, R.-P., Bungartz, H.-J., Niggel, A. und Rank, E. (2006). *Embedding, Organisation, and Control of Simulation Processes in an Octree-Based CSCW Framework*. In 11th International Conference on Computing in Civil and Building Engineering.

- [155] Myricom Inc. (2006). *Myrinet-2000*.
<http://www.myri.com/>
- [156] Nachtwey, B., Tölke, J. und Krafczyk, M. (2006). *Prototyp eines parallelen, interaktiven Rechenkerns zur hygro-thermischen Gebäudesimulation auf Basis der Lattice-Boltzmann Methode*. In BauSim. München, Germany.
- [157] Nell, J. (2001). *STEP on a Page*, NIST, Editor.
<http://www.mel.nist.gov/sc5/soap/>
- [158] Nemetschek AG (2006). *Nemetschek treibt Standardisierung von PDF in Architektur- und Ingenieurwesen voran*.
<http://nemetschek.de/pr>
- [159] Network Working Group (1985). *RFC 959: FILE TRANSFER PROTOCOL (FTP)*.
<http://tools.ietf.org/html/rfc959>
- [160] Network Working Group (1999). *RFC 2817: Hypertext Transfer Protocol - HTTP/1.1*.
<http://tools.ietf.org/html/rfc2817>
- [161] Neuberg, F. (2004). *Ein Softwarekonzept zur Internet-basierten Simulation des Ressourcenbedarfs von Bauwerken*. Dissertation, Lehrstuhl für Bauinformatik, Fakultät Bauingenieur- und Vermessungswesen, TU München, Germany.
- [162] O'Rourke, J. (2001). *Computational Geometry in C.*: Cambridge University Press.
- [163] OBERMEYER Planen + Beraten GmbH (2003).
<http://www.opb.de/>
- [164] Object Management Group (OMG) (2006). *Unified Modeling Language (UML), Version 2.0*.
<http://www.uml.org/>
- [165] Object Management Group Inc. (2004). *CORBA - Common Object Request Broker Architecture: Core Specification*, Version: 3.0.3.
<http://www.omg.org/docs/formal/04-03-01.pdf>
- [166] Olesen, B.W. und Dossi, F. (2005). *Neue Erkenntnisse über Regelung und Betrieb für die Betonkernaktivierung. Teil 1 u. Teil 2*. In HLH - Lüftung/Klima, Heizung/Sanitär, Gebäudetechnik. Vol: 56: S. 29-34 (Teil 1), 35-40 (Teil 2).
- [167] OMG Network Working Group (2006). *UUID/GUID Definition*.
<http://www.opengroup.org>

- [168] Open CASCADE S.A.S. (2006). *Open CASCADE Technology*, Version: 6.1.
<http://www.opencascade.org/org/doc/>
- [169] Petersen, M. (2002). *Ein Konzept verteilter Software-Komponenten zur Integration der thermischen Bauphysik in die Gebäudeplanung*. Dissertation, Institut für Numerische Methoden und Informatik im Bauwesen, TU Darmstadt, Germany.
- [170] Pilgrim, M., Bouchlaghem, D., Loveday, D. und Holmes, M. (2001). *A mixed reality system for building form and data representation*. In IEEE.
- [171] Poyet, P. et al. (1994). *ATLAS: Implementation of Knowledge Base Extensions*, in *ESPRIT III Project No. 7280*, E.C. ATLAS Deliverable D301a, Editor: Brussels.
- [172] Pültz, G. (2006). *Standardplanung versus Simulationen - wirtschaftlicher Nutzen moderner Planungswerkzeuge*. In BauSIM 2006. München, Germany.
- [173] Rangaraju, N. und Tek, M. (2001). *Framework for immersive visualization of building analysis data*. In IEEE.
- [174] Rauber und Rüger (2000). *Parallele und verteilte Programmierung*. Springer-Lehrbuch. Berlin/Heidelberg, Germany: Springer-Verlag.
- [175] RealVNC (2002). *VNC (Virtual Network Computing)*, Version: 4.1, Typ: Kommunikation.
<http://www.realvnc.com/>
- [176] Recknagel, Sprenger und Schramek (2005). *Taschenbuch für Heizung und Klimatechnik 2005/2006*. Vol. 72. München, Germany: Oldenbourg Industrieverlag.
- [177] Ritschel (1994/1995). *Raumklimatechnik, Band 1*: Springer Verlag, Berlin, Germany. S. 125-153.
- [178] Ruben, J. (2005). *Ein Multi-Agenten-System zur verteilten p-adaptiven Finite-Elemente-Simulation am Beispiel der Baugrund-Tragwerk-Interaktion*. Dissertation, Fachbereich Bauingenieurwesen und Geodäsie, TU Darmstadt, Germany.
- [179] Samet, H. (1984). *The Quadtree and Related Hierarchical Data Structures*. In *ACM Computing Surveys*. Vol: 16(2): S. 187-260.
- [180] Sarshar, M. und Christiansson, P. (2004). *Towards Virtual Prototyping In The Construction Industry: The Case Study Of The Diversity Project*. In *International Conference on Construction Information Technology*. Langkawi, Malaysia.
- [181] Scherer, R.J. (1995). *COMBI: Overview and Objectives, Product and Process Modelling in the Building Industry*. In Balkema, Rotterdam, the Netherlands. S. 503-510.

- [182] Scherer, R.J. (1996). *COMBI Final Report*, in *EU / CEC ESPRIT III Project No. 6609*: Dresden, Germany.
- [183] Scherer, R.J. (1996). *Objektorientierte Produktmodellierung im Bauwesen als Grundlage eines durchgängigen CAD/CAM-Konzeptes*. Arbeitsbericht über das Forschungsprogramm 694 der DFG. Bochum, Germany: D. Hartmann.
- [184] Schleicher und Blasinger (2004). *Regelungstechnik für den Praktiker*. Fulda, Germany: JUMO GmbH & Co. KG.
- [185] Schneider, S. (2006). *Interaktive massiv parallele Visualisierung großer Datenmengen aus Strömungssimulationen*. Dissertation, Fachbereich Informatik, TU Darmstadt, Germany.
- [186] Schulz, M., Krafczyk, M., Tölke, J. und Rank, E. (2001). *Parallelization Strategies and Efficiency of CFD Computations in complex geometries using Lattice-Boltzmann methods on High-Performance Computers*. In 3rd International FORTWIHR Conference on HPSEC. Erlangen, Germany: High-Performance Scientific and Engineering Computing.
- [187] SCI Institute - University of Utah (2004). *Scientific Computing, Scientific Visualization, Imaging*.
<http://www.sci.utah.edu/research/research.html>
- [188] Seifert, W. (2006). *HOAI. Honorartabellenbuch in Euro-Werten*. Werner, Neuwied.
- [189] Serror, M.H., Inoue, J., Adachi, Y. und Fujino, Y. (2006). *Building on IFC: E-Interaction with/within Structural Design Domain*. In Joint International Conference on Computing and Decision Making in Civil and Building Engineering. Montreal, Canada.
- [190] Skype Ltd. (2007). *Skype*, Version: 3.2.
<http://www.skype.com/intl/de/>
- [191] Sommer, W. (2004). *AutoCAD 2004 Kompendium, Professionelles Konstruieren*. Markt+Technik.
- [192] Spatial Corp. (2006). *3D ACIS Modeler*, Version: 16.
www.spatial.com/products/acis.html
- [193] SPI (2007). *Debian GNU/Linux*.
<http://www.debian.org/index.de.html>
- [194] ST-7 Extension Project (2005). *ST-7 Version 2.0 Release*.
<http://groups.yahoo.com/group/ST-7>

- [195] Strand, R.K., Liesen, R.J., Fisher, D.E. und Pedersen, C.O. (2002). *Modular HVAC Simulation and the Future Integration of Alternative Cooling Systems in a New Building Energy Simulation Program*. In ASHRAE Transactions. Vol: V. 108(2).
- [196] Sun Microsystems Inc. (2007). *Java Developer Network (JDN)*.
<http://java.sun.com/>
- [197] Sunday, D. (2007). *Intersections of rays, segments, planes and triangles in 3D*.
<http://geometryalgorithms.com/>
- [198] Sutherland, I.E. (1965). *The Ultimate Display*. In The 1965 IFIP Congress.
- [199] Tessmar Gebäudeleittechnik GmbH (2007). *Einsatzbereiche der Gebäudeleittechnik*.
<http://www.tessmar.de/>
- [200] Tölke, J. (2001). *Gitter-Boltzmann-Verfahren zur Simulation von Zweiphasenströmungen*. Dissertation, Lehrstuhl Bauinformatik, TU München, Germany.
- [201] Tölke, J. (2005). *A thermal model based on the lattice Boltzmann method for low Mach number compressible flows*. In Journal of Computational and Theoretical Nanoscience.
- [202] Tölke, J., Fahrig, T., Nachtwey, B. und Krafczyk, M. (2003). *Computational Steering in Civil Engineering*. In IKM Conference. Weimar, Germany.
- [203] Tölke, J., Krafczyk, M., Schulz, M. und Rank, E. (2000). *Discretization of the Boltzmann-Equation in velocity space using a Galerkin approach*. In Journal of Computer Physics Communications. Vol: 129: S. 91-99.
- [204] Touretzky, D.S. (1990). *Common Lisp: A Gentle Introduction to Symbolic Computation*: Benjamin/Cummings Publishing Company Inc.
<http://www.cs.cmu.edu/afs/cs.cmu.edu/user/dst/www/LispBook/book.pdf>
- [205] Treeck, C.v. (2004). *Gebäudemodell-basierte Simulation von Raumluftströmungen*. Dissertation, Lehrstuhl für Bauinformatik, TU München, Germany.
- [206] Treeck, C.v., Krafczyk, M., Kühner, S. und Rank, E. (2001). *Direct Building Energy Simulation Based on Large Eddy Techniques and Lattice Boltzmann Methods*. In Building Simulation 2001, 7th IBPSA Conference. Rio de Janeiro, Brazil.
- [207] Treeck, C.v., Krafczyk, M. und Schulz, M. (2001). *Lattice-Boltzmann Verfahren im Bauwesen, Tutorial eines grafisch-interaktiven Strömungslösers* In 13. Forum Bauinformatik. München: VDI-Verlag, Düsseldorf.
- [208] Trolltech (2004). *QSocket - QT*, Version: 3.3.
<http://doc.trolltech.com/3.3/qsocket.html>
- [209] Trolltech (2005). *Klassenbibliothek Qt*, Version: 3.31.

<http://www.trolltech.com/products/qt/index.html>

- [210] TROX GmbH (2007). *TROX-Technik Produkte*.
<http://trox.de/de/products/>
- [211] UNIQA Versicherungen AG (2004). *UNIQA Tower*.
<http://tower.uniq.at/>
- [212] Upson, C. et al. (1989). *The Application Visualization System: A Computational Environment for Scientific Visualization*. In IEEE Computer Graphics and Applications. Vol: 9(4): S. 30-42.
- [213] VDI Verlag (1996). *VDI 2078: Berechnung der Kühllast klimatisierter Räume (VDI-Kühllastregeln)*.
- [214] VISENSO GmbH (2005). *VISiT Projekt / basierend auf COVISE*.
<http://www.visenso.de/>
- [215] Wassermann, K. (1996). *Integration raum- und bauteilorientierter Daten in der Gebäudeplanung in einem zentralen Objektmodell*. Arbeitsbericht über das Forschungsprogramm 694 der DFG. Bochum, Germany: D. Hartmann.
- [216] Wellmann-Jelic, A. und Bilek, J. (2006). *Effizienzvergleich eines Agenten-Frameworks für parallele Ermüdungssimulationen mit klassischen Parallelisierungstechniken*. In Forum Bauinformatik. Weimar, Germany: Bauhaus-Universität.
- [217] Wooldridge, M. und Jennings, N.R. (1994). *Agent Theories, Architectures, and Languages: A Survey, Intelligent Agents*. In ECAI-94 Workshop on Agent Theories, Architecture, and Languages. Amsterdam, the Netherlands: Springer-Verlag.
- [218] World Wide Web Consortium (W3C) (1996). *XML - Extensible Markup Language*.
<http://www.w3.org/XML/>
- [219] Zhuang, L. (2006). *An Automatic Mesh Generator for a Computational Steering Framework*. Studienarbeit, CAB - Institut für Computeranwendungen im Bauingenieurwesen, TU Braunschweig, Germany.